

2

NPS 69-88-010

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A202 195



DTIC FILE COPY

THESIS

DTIC
ELECTE
04 JAN 1989
S D
& E

COMPARATIVE CONTROLLER DESIGN FOR A
MARINE GAS TURBINE PROPULSION PLANT

by

Vincent A. Stammetti

September 1988

Thesis Advisor:

David L. Smith

Approved for public release; distribution unlimited.

Prepared for:
Naval Postgraduate School
Monterey, California 93943-5000

89 1 04 020

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R. C. Austin
Superintendent

Harrison Shull
Provost

This report was prepared in conjunction with research conducted for the Naval Sea Systems Command and funded by the Naval Postgraduate School.

Reproduction of all or part of this report is authorized.

This report was prepared by:



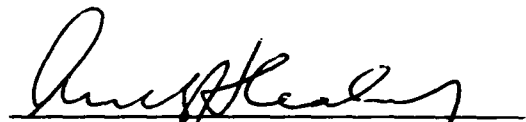
V.A. Stammetti



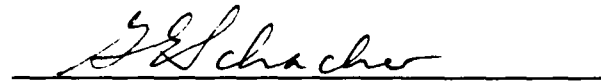
D. L. Smith
Assoc. Professor of Mechanical
Engineering

Reviewed by:

Released by:



Anthony J. Healey
Chairman, Department of
Mechanical Engineering



G. E. Schacher
Dean of Science and
Engineering

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NPS 69-88-010		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL (If applicable) Code 69	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000		7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Postgraduate School	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO	PROJECT NO
		TASK NO	WORK UNIT ACCESSION NO
11. TITLE (Include Security Classification) COMPARATIVE CONTROLLER DESIGN FOR A MARINE GAS TURBINE PROPULSION PLANT			
12. PERSONAL AUTHOR(S) Stammetti, Vincent A.			
13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED FROM TO	14. DATE OF REPORT (Year, Month, Day) 1988, September	15. PAGE COUNT 180
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		Marine Gas Turbine Control; Gas Turbine Modeling, theses. (MGM, E)	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Improved control of gas turbine propulsion plants could offer the Navy increased economic, maintenance, and tactical benefits. This thesis provides methods of steady state and dynamic computer modelling, as well as two non-proprietary control design methods. The classical proportional integral (PI) regulator design method and the modern linear quadratic regulator (LQR) controller design method are used to demonstrate a base for Navy redesign of existing gas turbine propulsion plant control systems. A comparison of the PI and LQR designs is conducted. In addition, a real or near-real time dynamic computer simulation is presented that has immediate application in the areas of model-based control and plant health monitoring.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Prof. David L. Smith		22b. TELEPHONE (Include Area Code) (408) 646-3383	22c. OFFICE SYMBOL Code 69Sm

Approved for public release; distribution is unlimited

Comparative Controller Design for a
Marine Gas Turbine Propulsion Plant

by

Vincent A. Stammetti
Lieutenant, United States Navy
B.S., United States Naval Academy, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the


NAVAL POSTGRADUATE SCHOOL
September 1988

Author:




Vincent A. Stammetti


Approved by:



David L. Smith, Thesis Advisor



A.J. Healey, Chairman
Department of Mechanical Engineering



Gordon E. Schacher,
Dean of Science and Engineering

ABSTRACT

Improved control of gas turbine propulsion plants could offer the Navy increased economic, maintenance, and tactical benefits. This thesis provides methods of steady state and dynamic computer modelling, as well as two non-proprietary control design methods. The classical proportional integral (PI) regulator design method and the modern linear quadratic regulator (LQR) controller design method are used to demonstrate a base for Navy redesign of existing gas turbine propulsion plant control systems. A comparison of the PI and LQR designs is conducted. In addition, a real or near-real time dynamic computer simulation is presented that has immediate application in the areas of model-based control and plant health monitoring.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



TABLE OF CONTENTS

I.	INTRODUCTION -----	1
	A. CONTROLLER BACKGROUND -----	3
II.	PREVIOUS GAS TURBINE MODELLING AND CONTROL -----	7
	A. EARLY COMPUTER MODELS -----	7
	B. DYNAMIC COMPUTER MODELS FOR MARINE ENGINE SIMULATION -----	8
	C. RECENT CONTROL DESIGN TECHNIQUES -----	12
III.	PREVIOUS WORK AT NPS -----	16
IV.	STEADY STATE SOLUTION -----	23
V.	DYNAMIC SIMULATION -----	36
VI.	CONTROLLER DESIGN AND COMPARISON -----	43
VII.	CONCLUSIONS AND RECOMMENDATIONS -----	58
	APPENDIX A: VARIABLE RANGE DETERMINING PROGRAM -----	60
	APPENDIX B: SOLUTION CONVERGENCE PROGRAM -----	68
	APPENDIX C: STEADY STATE VARIABLE AND PARTIAL DERIVATIVE PROGRAM -----	78
	APPENDIX D: SUBROUTINES USED -----	88
	APPENDIX E: DYNAMIC SIMULATION PROGRAM -----	143
	APPENDIX F: COMPARISON OF "A" MATRIX COEFFICIENTS ----	152
	APPENDIX G: SIMULATION VS. DATA RUNS -----	159
	LIST OF REFERENCES -----	166
	INITIAL DISTRIBUTION LIST -----	168

LIST OF TABLES

1.	STEADY STATE CONVERGENCE MAP "A" MATRIX COEFFICIENT SIGN ERRORS -----	37
2.	"A" MATRIX COEFFICIENT CURVE FITS -----	40
3.	CONTROLLER DESIGN COMPARISON -----	57
4.	A11 MATRIX COEFFICIENT CORRELATION DATA -----	153
5.	A11 MATRIX COEFFICIENT CORRELATION DATA -----	154
6.	A13 MATRIX COEFFICIENT CORRELATION DATA -----	155
7.	A21 MATRIX COEFFICIENT CORRELATION DATA -----	156
8.	A22 MATRIX COEFFICIENT CORRELATION DATA -----	157
9.	A23 MATRIX COEFFICIENT CORRELATION DATA -----	158

LIST OF FIGURES

1.	Typical ITC Control Scheme -----	2
2.	F100 Control Scheme -----	15
3.	NPS Gas Turbine Propulsion Plant Emulator -----	17
4.	Herda's Multiport Diagram -----	21
5.	Steady State Convergence -----	25
6.	Example of Crossings Exhibited by Computed and Guessed Values of P4 -----	27
7.	Flowchart of VRD Algorithm -----	29
8.	Steady State Convergence Map -----	34
9.	Smoothed Dynamic Transition Map -----	38
10.	Effect of Linearizations on NS -----	41
11.	Author's PI Controller--Plant Configuration -----	45
12.	Proportional Control--Deceleration -----	46
13.	Proportional Control--Acceleration -----	47
14.	Proportional Integral Control--NG, Deceleration --	49
15.	Proportional Integral Control--NS, Deceleration --	50
16.	Controller Comparison--NG, Deceleration -----	52
17.	Controller Comparison--NS, Deceleration -----	53
18.	Controller Comparison--NG, Acceleration -----	54
19.	Controller Comparison--NS, Acceleration -----	55
20.	Simulation and Validation for Run 1 Data, NG -----	160
21.	Simulation and Validation for Run 1 Data, NS -----	161
22.	Simulation and Validation for Run 2 Data, NG -----	162
23.	Simulation and Validation for Run 2 Data, NS -----	163

24.	Simulation and Validation for Run 3 Data, NG -----	164
25.	Simulation and Validation for Run 3 Data, NS -----	165

LIST OF SYMBOLS AND ABBREVIATIONS

E	= Fuel Energy Realized at HP Turbine
JD	= Dynamometer Inertia
JG	= Gas Generator Inertia
MA	= Air Mass Flow Rate
MAF	= Combined Fuel and Air Mass Flowrate
MF	= Fuel Mass Flow Rate
NG	= Gas Generator Speed
NS	= Power Turbine/Dynamometer Speed
P2	= Compressor Discharge Pressure
P2C	= Computed Compressor Discharge Pressure
P2G	= Guessed Compressor Discharge Pressure
P4	= High Pressure Turbine Discharge Pressure
P4C	= Computed High Pressure Turbine Discharge Pressure
P4G	= Guessed High Pressure Turbine Discharge Pressure
QC	= Compressor Torque
QD	= Dynamometer Torque
QFPT	= Free Power Turbine Torque
QHPT	= High Pressure Turbine Torque
T2	= Compressor Discharge Temperature
T4	= High Pressure Turbine Discharge Temperature
WW	= Dynamometer Water Weight

ACKNOWLEDGMENTS

The author would like to express his thanks to Professor David Smith, whose unbounded enthusiasm, patience, and unyielding standards of excellence provided much of the inspiration and motivation to accomplish this work. The author would also like to express his unending gratitude to his wife and son for their many sacrifices, understanding, and devotion throughout the master's degree evolution.

I. INTRODUCTION

Many modern surface Naval marine propulsion plants are a combination of gas turbine engines with controllable reversible pitch propellers. This presents the problem of matching the engine RPM to the most efficient pitch which has been accomplished through the use of an integrated throttle control (ITC). Figure 1 shows a block diagram of a typical ITC control scheme.

The implementation technology for Figure 1 is well over 20 years old and its limitations are now well defined. Today, technology exists that will allow the antiquated analog mechanisms and current computerized systems to be replaced by smaller more reliable digital controls and hardware. This approach suggests that the following benefits could be realized:

- (1) Reduction of maintenance "nightmares" that develop due to the intricacy and number of small parts in components such as mechanical fuel governors;
- (2) More reliable and compact circuitry would modify present hardware such as the Free Standing Electronic Enclosure, propulsion and electrical control consoles, and current engine health monitoring equipment;
- (3) Advances in the ability to model and simulate gas turbine performance would allow plant performance to be significantly improved, thereby increasing plant efficiency and translating to lower operating costs;
- (4) New techniques in engine health monitoring and analysis provide essential real time data on plant performance to the operators, allowing better and

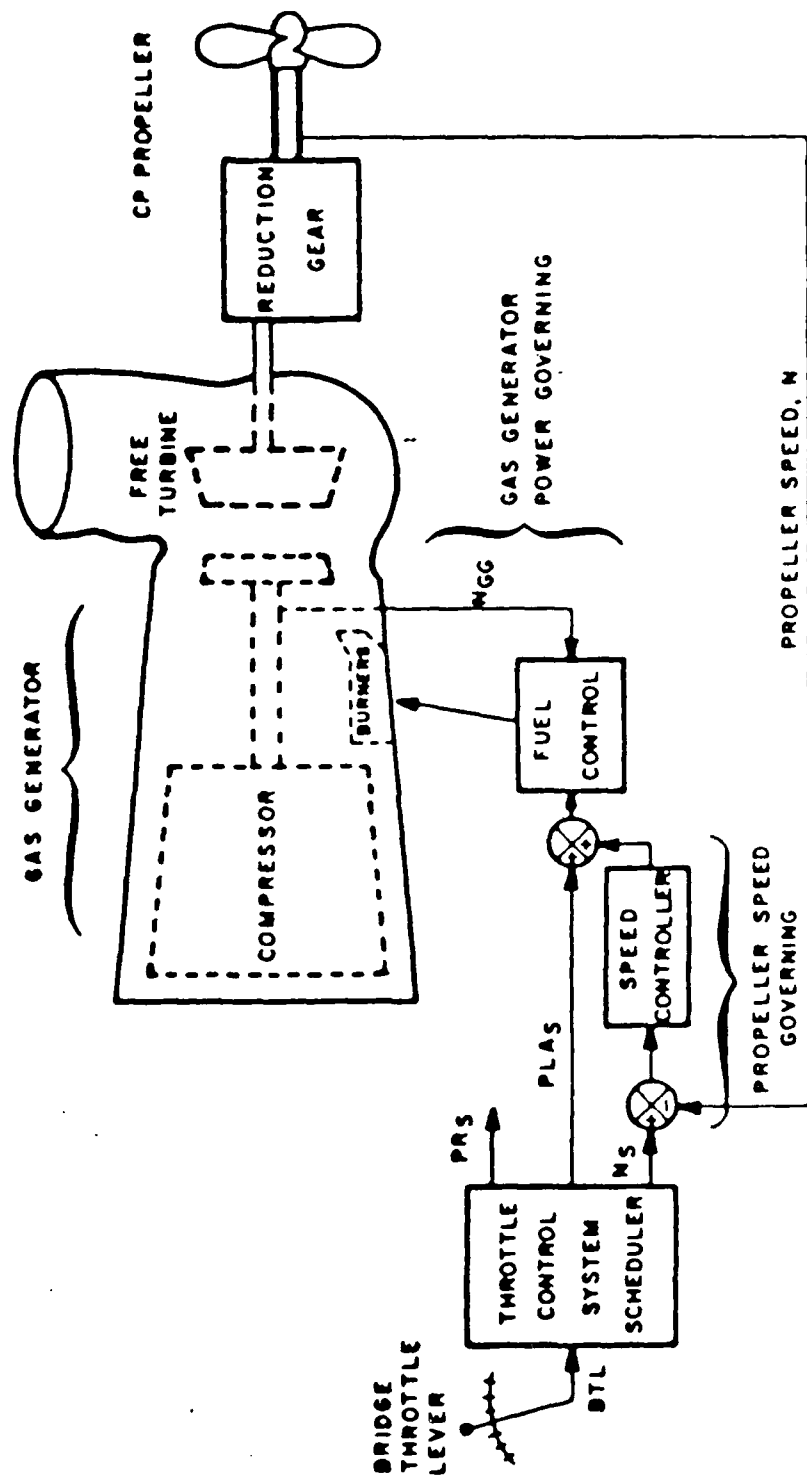


Figure 1. Typical ITC Control Scheme

more rapid evaluation and response to a potential or actual engineering casualty;

- (5) More compatibility between control systems could be achieved, thereby reducing the number of different repair parts that must be stocked in the naval supply system. More commonality would also streamline the training process of personnel responsible for maintaining and operating the systems;
- (6) Inherent flexibility through reprogramming of computer-based controls would pave the way for future developments.

A. CONTROLLER BACKGROUND

During the late seventies and early eighties the marine gas turbine industry hotly debated the pros and cons of analog vs. digital control to implement integrated throttle control [Ref. 1]. The advocates of analog control were of the opinion that this technology was reliable and could perform all necessary calculations required for effective plant control. It was felt that little would be gained in the way of reduction of component count or system reliability through digital systems. This thought process led Rolls Royce to choose analog systems for warship controls, and led General Electric to a similar conclusion for the main fuel control on the LM-2500.

The digital advocates on the other hand, had the foresight to realize that advances in technology would be more easily implemented in a digital base, and that reliability would indeed be as good, if not better than, analog systems. With the advent of the microprocessor, the component count can indeed be reduced with a carefully

executed design process. This was demonstrated by the aviation community first on the F100 engine [Ref. 2]. A natural progression would be for the marine gas turbine community to follow suit. It must be realized that some analog fuel system control components will probably always be required, particularly in the sensing and actuation areas.

Perhaps the most compelling reason today to convert to digital control is the advent of intelligent control. In this approach, it is possible to control a large quantity of measured and unmeasured variables with a limited amount of operator intervention to meet the dynamic needs of the operator.

Typically, a good control design approach consists of eleven steps. These steps contain three "feedback loops" which provide the means for modification or improvement should the designer desire. This control design approach is as follows:

- (1) Specifications for control design;
- (2) Evaluation of plant function;
- (3) Plant mathematical modeling;
- (4) Plant model validation--open loop simulation;
- (5) Selection of control strategy;
- (6) Selection of actuators, sensors;
- (7) Dynamic modeling of actuators, sensors;
- (8) Selection of controller action;

- (9) Theoretical controller design;
- (10) Controller validation--closed loop simulation;
- (11) Prototype.

The design feedback loops exist between steps 4 and 3, between steps 10 and 8, and between steps 10 and 5. Utilization of computer-aided design techniques in the design, validation, and optimization of control schemes provides an efficient and economical method for selecting the most suitable candidate for hardware development. Prudent selection of designs is essential considering the complexity and large capital expenditure incurred as the design progresses from the conceptual phase to its final form. Inherent in this approach is the need for evaluation and modelling of gas turbine performance (step 3). Consequently, while this thesis is dealing with marine gas turbines, much early work was done in the area of aviation gas turbine modeling and control. We begin with a review of these efforts. Chapter II is a review of previous recent work in gas turbine modelling and control. Chapter III is a review of work previously performed at the Naval Postgraduate School. Chapters IV and V detail the steady state and dynamic simulations of this work. Chapter VI contains the design methods for a classical proportional integral (PI) regulator and a modern linear quadratic (LQR) regulator. Also in Chapter VI the controller designs are compared for

operation in a simulated sea state. Conclusions and recommendations make up Chapter VII.

II. PREVIOUS GAS TURBINE MODELLING AND CONTROL

A. EARLY COMPUTER MODELS

Gas turbines in use today for marine propulsion are for the most part derivatives of aviation gas turbine engines that have been "marinized" for use at sea. As one would expect, several computer simulations were developed to evaluate and predict system performance. The early simulations were developed by the aviation industry and provided a substantial data base for development of more advanced computer models. A short summary of some of the major early aircraft simulations is given below [Ref. 3]:

- (1) SMOTE--Developed in 1967 by the Turbine Engine Division of the U.S. Air Force Propulsion Laboratory (AFAPL), Wright-Patterson AFB, Ohio. It is capable of calculating steady-state design and off design performance of a two-spool turbofan engine.
- (2) GENENG--Developed in 1972 by NASA's Lewis Research Center (LeRC), Cleveland, Ohio. Its purpose is to improve the versatility of SMOTE. Steady-state design and off-design performance of one- and two-spool turbojets can be calculated as well as the two-spool turbofan.
- (3) GENENG II--Derivative of GENENG, it calculates steady-state performance of two- or three-spool turbofan engines with as many as three nozzles.
- (4) NEPCOMP--Developed in 1974 by the Naval Air Development Center (NADC), Warminster, Pennsylvania. The flexibility inherent to NEPCOMP allows for calculation of steady-state performance of gas-turbine engines with multiple spools, including turbojets, turbofans, turboshafts, and ramjets.
- (5) DYNGEN--Developed in 1975 by LeRC, it combined the capabilities of GENENG and GENENG II for calculating

steady-state performance of gas turbine engines with multiple spools. The additional capability of calculating transient performance was also added.

- (6) NNEP--Jointly developed in 1975 by NASA, LeRC, and NADC. This computer code is able to simulate steady-state design and off design performance of almost any conceivable gas turbine engine simulation.

As can be seen above, the majority of the early work was devoted to steady-state simulations. A major shortfall was a lack of dynamic simulation capability. At this point it is prudent to shift the emphasis from the work performed by the aviation industry and concentrate on the contributions made in the marine gas turbine industry in the area of dynamic simulation.

B. DYNAMIC COMPUTER MODELS FOR MARINE ENGINE SIMULATION

Engineers at David W. Taylor developed equations to mathematically model various engine components for a "building block" approach to modelling [Ref. 3]. Once these were established, a system of common component interface locations was defined and the locations were numbered. Equations were then developed for the numbered major gas turbine components, including compressors, burners, turbines, and engine load. Dynamic equations were then developed to describe speed, power balances, mass accumulation, and energy accumulation.¹ An iterative approach was then utilized to balance the performance

¹Information used for this portion of the discussion only relates to a simulation of a single spool engine configuration.

characteristics of the various engine components. A Newton-Raphson technique was used to achieve convergence. The results of the simulations yielded good comparisons between the manufacturer's simulation and the existing experimental data.

Beginning in the early seventies, the U.S. Navy initiated The Gas Turbine Ship Propulsion Control Systems Research and Development Program [Ref. 4]. The Navy chose Propulsion Dynamics, Incorporated to conduct the program which was designed to develop a machinery dynamics and control system data base. The program involved computer simulations of total propulsion systems, which were validated by shipboard and model testing. The program continued into the eighties and was still generating technical papers as recently as 1986. The program was successful in developing a theoretical design base for gas turbine propulsion systems. Major conclusions were drawn in the following areas [Ref. 5]:

- (1) Propulsion systems cycling;
- (2) Propeller speed governing;
- (3) Gas generator power governing;
- (4) Combined Power and Speed Governing.

Based on data obtained during the program, a ship propulsion control system was devised for use in computer simulations. The control system was of the classical integral variety, whose gains were fixed via a "cut and try"

method. Linear controller gains were obtained for various wave conditions and engine speeds, then tabulated and compared. In a current application the gain is fine tuned via the "sea state adjust" control found on the propulsion control consoles aboard DD-963 class destroyers to account for variations in the load and non-linear propulsion plant. Figure 1 shows a block diagram of the ship propulsion control system used. Simulations of this approach tended to give good results when compared to model and ship generated data [Ref. 5]. However, the approach generated some interesting observations regarding a gas turbine engineering plant's response to seaway- and maneuver-induced unsteady loading, which are indeed confirmed by the experience of the author who served as Main Propulsion Assistant aboard a DD-963 class destroyer. In high seas, gas turbine plants experience a good deal of engine/propeller cycling due to constant changes in propeller loading as the ship moves through the water. A ship configured with two propulsion shafts experiences a good deal of propeller load variation during turns, particularly during high speed turns. Naturally, these conditions cause numerous changes in engine speed, resulting in engine wear and potential overspeeding of the engine gas generator should the propulsion load be lost for some reason. It should be noted at this point that these two phenomena can be thought of as "disturbances" to the plant.

Returning to general control development, modern control theory provided the next logical step in controller design. In this work, state space techniques applied to gas turbines have yielded positive results. Such state variable methods allow the control system designer to gain an understanding of the inherent input cross-channel coupling dynamic characteristics of the system and to take advantage of coupling which exists between input and output variables.

In the late seventies students and faculty at the Naval Postgraduate school applied state space techniques to a linearized model of an FFG-7 ship propulsion system [Ref. 6]. Dynamic propulsion system equations were developed for the FFG-7 and then linearized, the appropriate matrices developed, and the dynamic simulations conducted. The results demonstrated that the linear model described the system behavior reasonably well.

Another mathematical model was developed at Tsinghua University, Beijing, China in the mid-eighties [Ref. 7]. A three shaft marine gas turbine was modelled and simulated using state space techniques, and two different numerical methods were used to obtain convergence. The convergence methods used were: (1) the varying coefficient method, and (2) the small deviation method. The difference in methods lies in the fact that only small system perturbations can be considered in the latter, while large perturbations can be considered in the former. In the first method the initial

point of linearization lies in the unsteady regime. The real beauty of the varying coefficient method is that transients under large perturbations can be obtained with sufficient accuracy using linearized equations. Results from the two simulation techniques were compared and the varying coefficient method was deemed more accurate.

C. RECENT CONTROL DESIGN TECHNIQUES

There are numerous methods by which one can design a modern controller for an automatic system. When a state space approach is taken to design, there are basically two ways to approach the task: (1) The Pole Placement method, and (2) The Linear Quadratic Regulator technique (LQR). The Pole Placement method requires that the location of the desired system closed loop poles be known. Since the optimum closed loop poles of a system may not be known during design, the LQR method is often a better choice. The LQR method optimizes the design of the controller, based on the inputs of various matrices and a cost function. The LQR controller often requires an observer to calculate the states, it then calculates the error between actual and desired states and computes the gains such that stability is guaranteed and the integrated error minimized. (The theory of this approach will be reviewed in more detail in the following chapters).

Kidd, Munro, and Winterbone examined the potential of a digital control scheme designed using LQR state space

techniques [Ref. 8]. The plant model was one of a two-shaft, two-turbine vessel with a combination of a sprint and a cruise turbine on each shaft coupled to a controllable reversible pitch propeller via a reduction gear. The simulations were performed using a FORTRAN IV digital, non-linear, dynamic computer simulation which included steady state data for the non-linear propeller and thrust characteristics. A digital controller was developed using state space techniques, eventually culminating in a gain-scheduled multivariable controller which was constructed from a selection of linear compensator designs. For comparison purposes a conventional control system was designed as a yardstick by which to measure the digital control system. Both controllers were then implemented in the non-linear ship simulation model. The responses of the two controllers were compared for several maneuvers and the multivariable controller demonstrated a much faster speed of response and less overshoot on propeller-shaft torque output. The multivariable controller constrained the propeller well within safe and acceptable operating limits. The improvements in response of the propulsion plant improved the ship speed response which resulted in ship acceleration and stopping time improvements, i.e ship maneuverability improvements.

LQR controllers have also been designed for the F-401 and F100 aerospace turbofan engines. Figure 2 is a block

diagram of the F100 control model [Ref. 2]. Similar research was done to apply LQR techniques to the design of a power turbine governor for a turboshaft engine driving a helicopter rotor blade [Ref. 9]. In that work, a GE-700 turboshaft engine was modelled using state space methods and was mathematically coupled to a linear lumped capacitance model of an articulated rotor blade. The two were then combined into an overall system matrix and simulated; the results were compared to a conventional governor's performance. The performance was increased in the areas of time response and overshoot in power turbine speed. These results seem to parallel the results obtained by Kidd, Munro, and Winterbone, but for a different application.

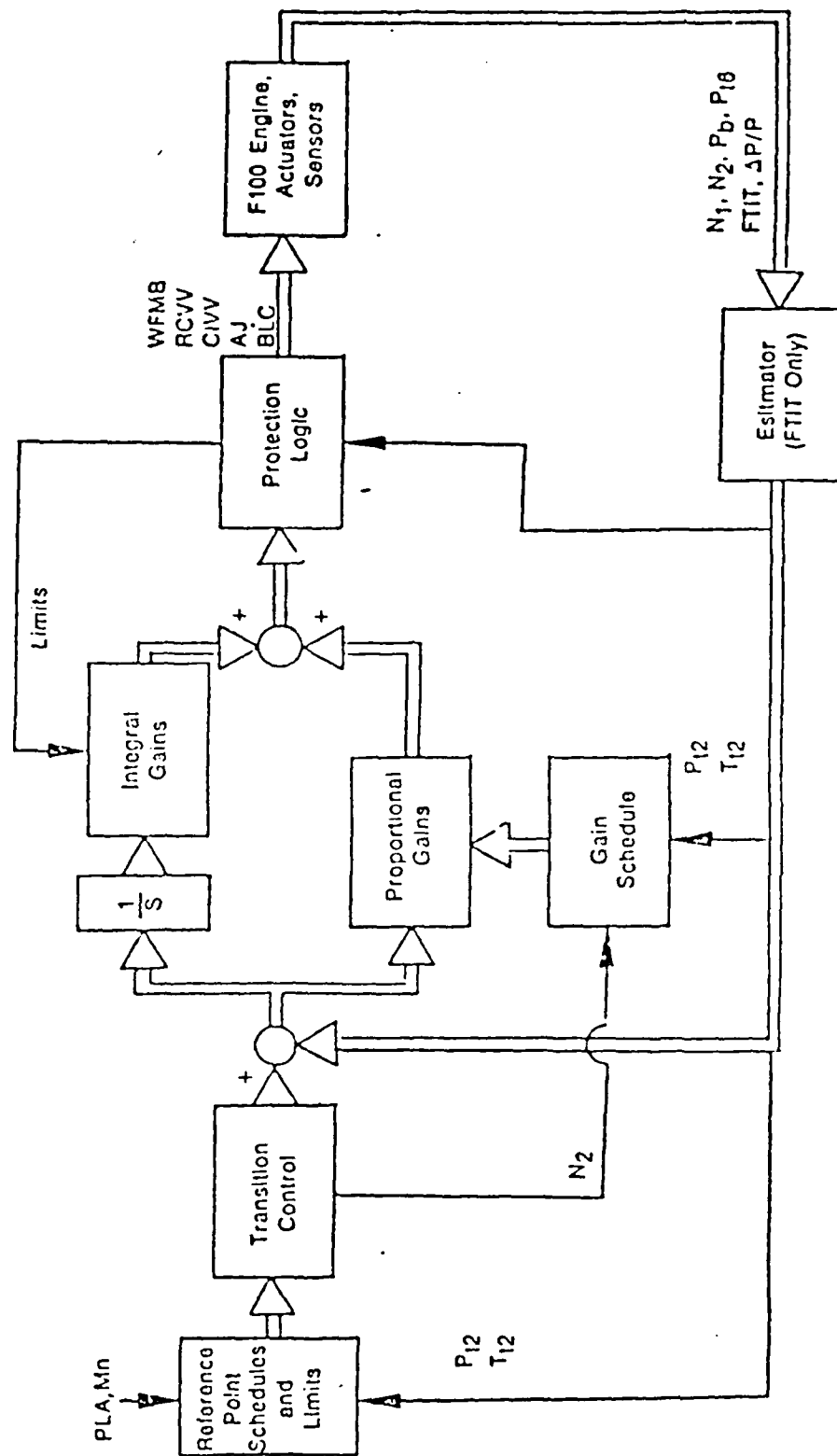


Figure 2. F100 Control Scheme

III. PREVIOUS WORK AT NPS

The plant being considered here is a Boeing Model 502-6A 175 horsepower gas turbine connected to a Clayton 17-300 water brake dynamometer as shown in Figure 3. The gas turbine is divided into two separate sections, a gas generator section and a power output section. The gas generator is composed of a single entry, single stage centrifugal compressor connected to a single stage axial flow high pressure turbine (HPT). Two cross-connected through-flow type combustion chambers provide an aerodynamic coupling between the turbine and compressor. An accessory drive section is geared off the gas generator shaft, and contains the electric starter, tachometer generator, fuel pump, governor, and lube oil pump. The power output section consists of an axial flow free power turbine (FPT), reduction gearing, and output shaft. The gas generator and power output sections are connected aerodynamically.

Previous work by Johnson [Ref. 10] (hardware design and implementation), Herda [Ref. 11] (computer modelling and simulation), and Miller [Ref. 12] (model testing and modification), has provided the starting point for the present work.

The state space model previously developed has been slightly modified, but remains essentially the same with the

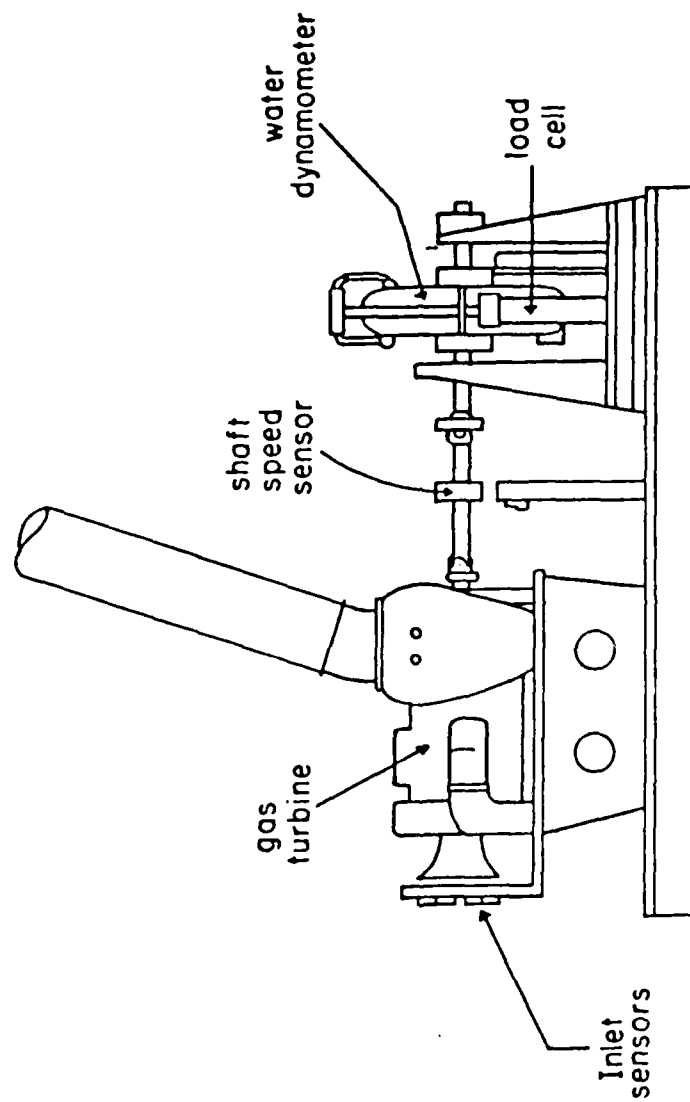


Figure 3. NPS Gas Turbine Propulsion Plant Emulator

exception of one of the plant inputs. The model for the present work applies the state space linearization:

$$\dot{\tilde{x}} = \tilde{A} \tilde{x} + \tilde{B} \tilde{u} \quad (1)$$

where

\tilde{x} = the state vector,
 \tilde{u} = the input vector,
 \tilde{A} = the state coefficient matrix,
 \tilde{B} = The input coefficient matrix.

The states are defined as gas generator speed (NG), power turbine/dynamometer speed (NS), and mechanical energy resulting from fuel combustion (E). The plant inputs are fuel flow rate (MF) and dynamometer torque (QD).

Perturbations which are the basis for the linearizations are accomplished by using the equation:

$$X = X_0 + x \quad (2)$$

where

X_0 = the initial value,
 $x = \delta x$ = the perturbation from the initial value,
 X = the current value.

All plant state space variables are represented in this manner. Employing the perturbational variables, the state space equation becomes:

$$\begin{Bmatrix} \dot{ng} \\ \dot{ns} \\ \dot{e} \end{Bmatrix} = A \begin{Bmatrix} ng \\ ns \\ e \end{Bmatrix} + B \begin{Bmatrix} mf \\ qd \end{Bmatrix}$$

The elements of the "A" and "B" matrices are calculated using Taylor series expansions on each plant component, retaining only first order terms. So, the elements of the state space "A" and "B" matrices can be written symbolically as:

$$\begin{array}{lll} a_{11} = \partial Ng / \partial ng & a_{12} = \partial Ng / \partial ns & a_{13} = \partial Ng / \partial e \\ a_{21} = \partial Ns / \partial ng & a_{22} = \partial Ns / \partial ns & a_{23} = \partial Ns / \partial e \\ a_{31} = \partial E / \partial ng & a_{32} = \partial E / \partial ns & a_{33} = \partial E / \partial e \\ \\ b_{11} = \partial Ng / \partial mf & b_{12} = \partial Ng / \partial qd & \\ b_{21} = \partial Ns / \partial mf & b_{22} = \partial Ns / \partial qd & \\ b_{31} = \partial E / \partial mf & b_{32} = \partial E / \partial qd & \end{array}$$

Herda developed both steady state and dynamic computer simulations to describe the behavior of the plant. The dynamic equations were derived from quadratic curve fits of steady state data collected during operation of the gas turbine/dynamometer. Uncorrected variables were used, primarily because the conditions in the test cell remained near standard conditions at all times. The modifications to the computer code to correct for temperature and pressure

could easily be added after successful concept validation. The cause and effect multiport model used for that work is depicted by Figure 4. It was initially proposed by Johnson, then expanded by Herda.

It is apparent from the multiport model that the variable coupling the gas generator and power output sections was the pressure P_4 . P_4 is both the high pressure turbine exhaust pressure and the free power turbine inlet pressure. Herda's steady state model was developed on the premise that for any operating point there was one fuel flow rate (MF), one high pressure turbine inlet pressure (P_2), and one high pressure turbine exhaust/free power turbine inlet pressure (P_4). Inputs to the model were gas generator speed (NG) and dynamometer speed (NS). From these inputs an initial fuel guess was computed. Convergence to the correct P_2 and P_4 was then attempted using a modified Newton-Raphson algorithm. If the pressures could not be converged, the fuel estimate was modified using a golden section technique. These iterations continued until either satisfactory convergence to specified criteria was obtained (in which case a torque comparison between the compressor and high pressure turbine was performed) or convergence failed and no solution was obtained. If the torque comparison failed to meet its convergence criteria, the iterations continued as just described. If satisfactory convergences between the pressures and torques were obtained, the routine calculated

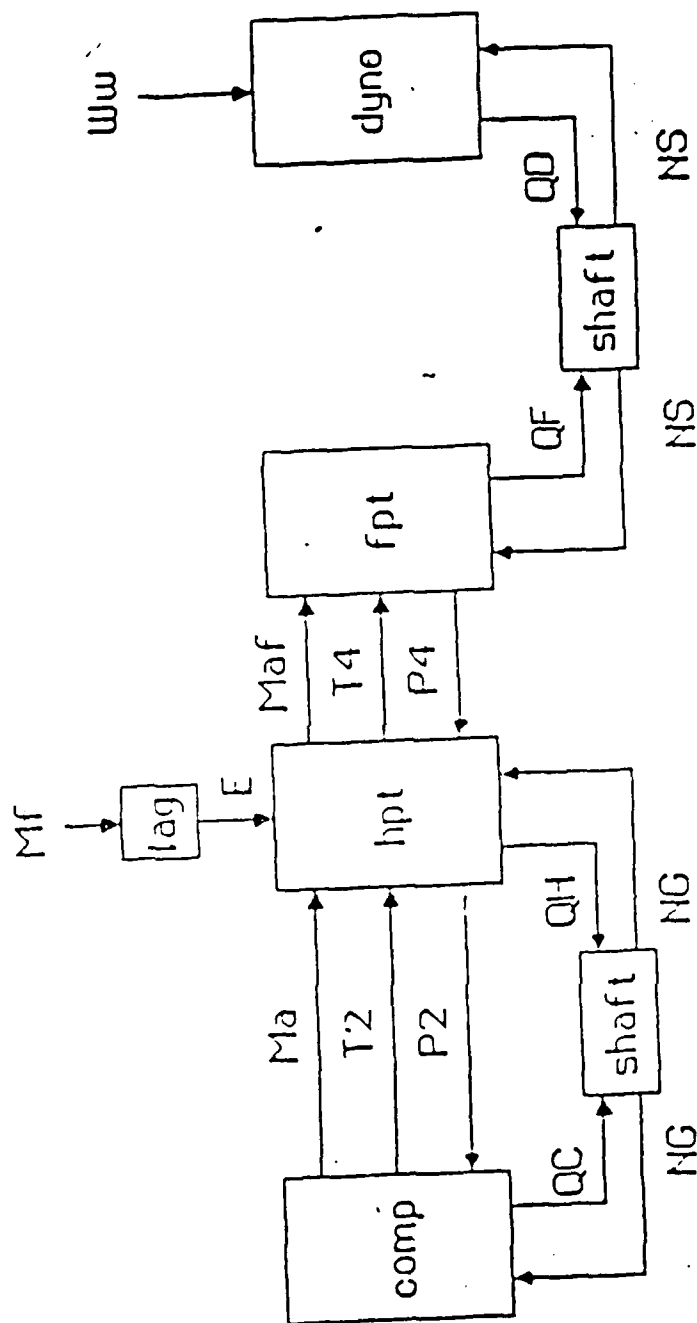


Figure 4. Herda's Multipoint Diagram

the remaining plant variables and the state space "A" and "B" matrices.

Herda performed limited testing of his steady state model and was able to obtain some good results, but he did experience numerical instability and failed convergence in some instances. Herda's dynamic model was developed using mainframe dynamic simulation software.

Miller's work focused on solving the numerical instability problem encountered by Herda. A performance envelope was developed and additional data obtained for analysis. Miller made minor modifications to the model and investigated alternative convergence methods with some success, but he also encountered numerical instabilities.

A summary of previous work is as follows:

- (1) The cause-effect multi-port model worked well to portray system response;
- (2) Computer algorithms derived from the multi-port model provide a method for linearization and state space matrix computation based upon steady state experimental data;
- (3) Numerical convergence for the steady state model was uncertain for portions of the plant operating envelope; and,
- (4) Great accuracy was required in the computation of steady state plant variables, particularly the pressure P4.

IV. STEADY STATE SOLUTION

The ability to obtain a steady state solution at any point in the operating envelope was prerequisite to the later dynamic simulation and controller design phases of this work. The criteria for an acceptable steady state solution was to achieve a torque balance between the compressor and high pressure turbine. A zero torque differential is indicative of gas generator balance and steady state operation, while a non-zero torque differential is indicative of gas generator acceleration or deceleration.

As in Herda's solution method, there was assumed to be a distinct MF, P2, and P4 for every steady state torque value, and these quantities were required for calculation of the remaining plant variables.

Techniques previously used for converging torque and pressure values were slope methods. Several other mathematically intense slope convergence algorithms were initially investigated in the present work, these also proved to be inadequate. To better visualize the behavior of the torque and pressure curves being dealt with, a torque balance equation was derived using the compressor and high pressure turbine equations developed by Herda. The resulting equation was a quadratic expression in terms of five variables: MF, P2, P4, NG, and NS. Three of these

(MF, NG, and NS) were known or could be calculated for a particular operating point, leaving P2 and P4 as unknowns. A grid procedure was employed to solve for P2 and P4 which forced two criteria to be met:

- (1) The gas turbine must be torque balanced;
- (2) All component input-output relationships must be satisfied.

The modified multiport diagram of Figure 5 illustrates the process.

Fuel (MF) and a range of potential P2G values (P2 "guessed") were used as inputs. A corresponding value of P4G was calculated from the torque balance quadratic equation. An imaginary root check discarded any imaginary roots, leaving only the real roots for consideration as possible solutions. Roots acquired using the negative radical portion of the quadratic equation were defined as "low energy" solutions, while roots acquired using the positive radical were defined as "high energy" solutions. It was decided that should the situation arise where both high and low energy solutions existed in P4G, the low energy solution would be chosen because it corresponded physically to less fuel used for the operating point. Each pair of guessed pressures (P4G) was then input into calculations to check for torque balance. If torque balance was achieved, the corresponding values were recorded and the routine continued. If the torque balance checks failed, the next value of P2G was input and the process repeated. Torque

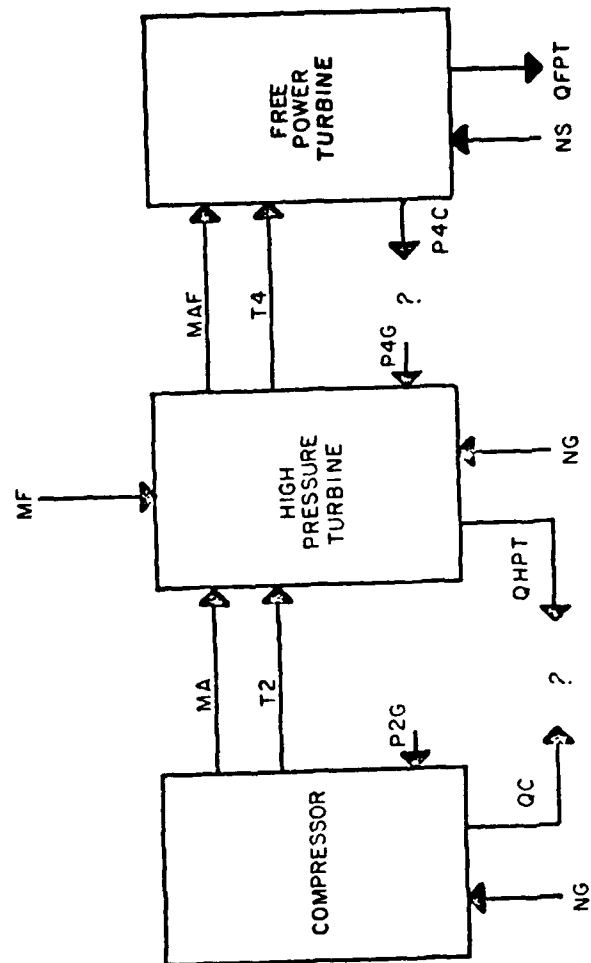


Figure 5. Steady State Convergence

balance caused the computation of P4C, the computed P4 pressure, which was calculated from a subroutine involving component input output relationships with P2G and MF as the inputs. Crossing logic was then used to detect points where the P4C and P4G curves met (or crossed). A crossing of these curves was considered to be a potential operating point of the plant. Figure 6 shows an example of P4C and P4G curves crossing.

The results of this procedure fell into one of the following categories:

- (1) A solution existed, but outside of the valid P2 range;
- (2) Multiple crossings existed in the valid P2 range;
- (3) Imaginary solutions existed;
- (4) A combination of 1, 2, and 3;
- (5) Only one solution existed in the valid P2 range;
- (6) No solution existed (no crossings).

Convergence to a root in category 1 or 3 above could result in an incorrect steady state solution. This explains some of the numerical instability and inability to converge some points in the operating envelope encountered in the past.

The existence of multiple roots presents the rather formidable task of consistently extracting the root that leads to the correct steady state solution. In the case of two roots in the valid P2 range, it was discovered that

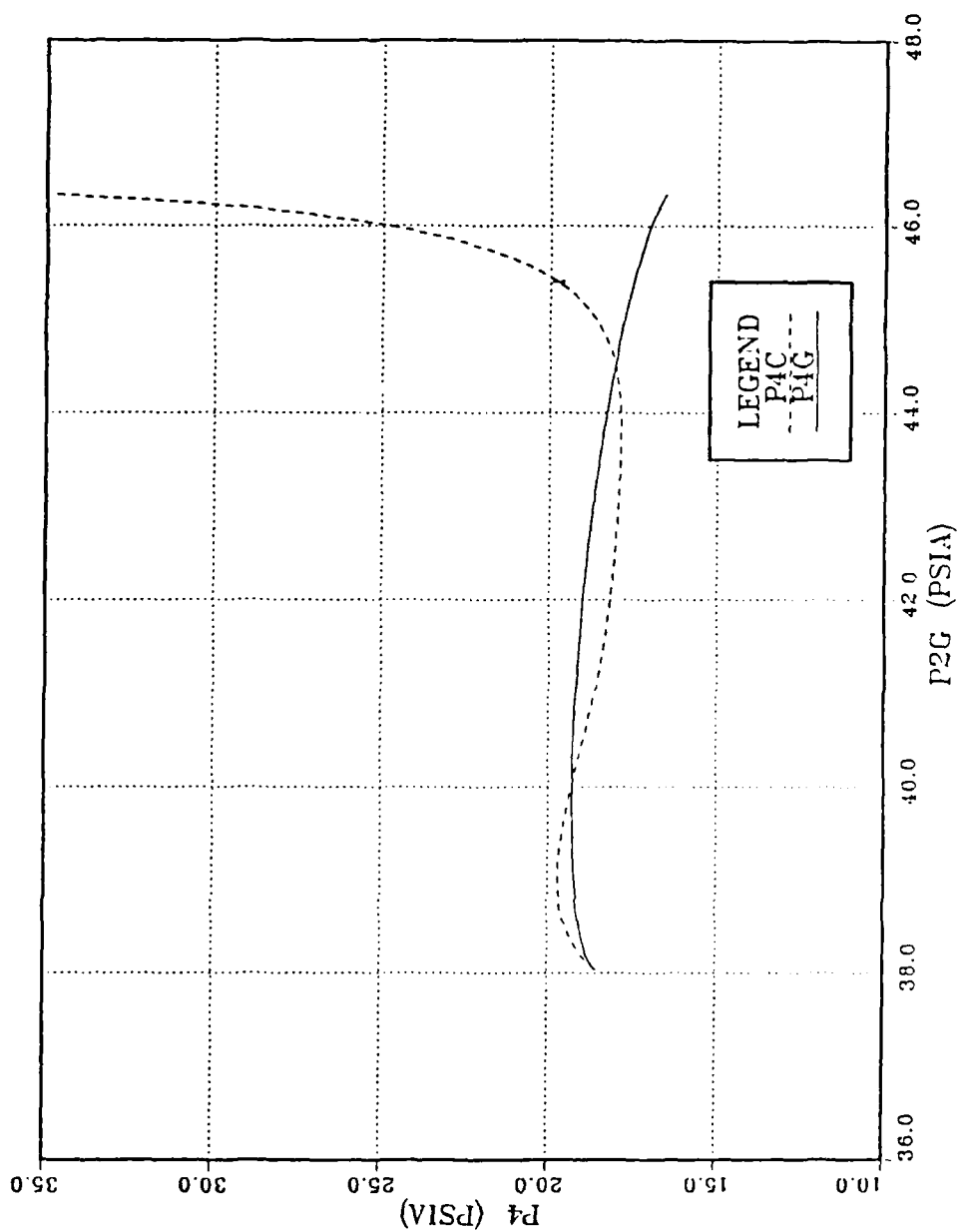


Figure 6. Example of Crossings Exhibited by Computed and Guessed Values of P_4

often the high energy solution provided a better steady state answer than the low energy solution.

Eventually this method was abandoned because of inaccuracies in equation coefficients and the large changes in the "A" and "B" matrices which occurred for very small changes in P4. For this reason a grid search technique was adopted. Although computationally intense, the method considered all possible combinations of MF, P2, and P4 within specified ranges, thereby eliminating the problem of converging on the first root which occurs in gradient methods.

Two computer algorithms were developed, one to provide the MF, P2, and P4 ranges to be searched, and the second to converge these three values. Figure 7 is a flowchart for the first algorithm which is called the Variable Range Determining (VRD) algorithm, a copy of which is included as Appendix A. The inputs to the program were gas generator speed (NG), power turbine speed (NS), and the number of iterations for each of the variables: MF, P2, and P4. This number of iterations determined both the size of the search increments and the width of search area. The fuel initial guess was computed by subroutine NGNSMF to determine the starting point for fuel iteration. A copy of NGNSMF and all other subroutines used is included as Appendix D. The variable initialization section was then entered to set the

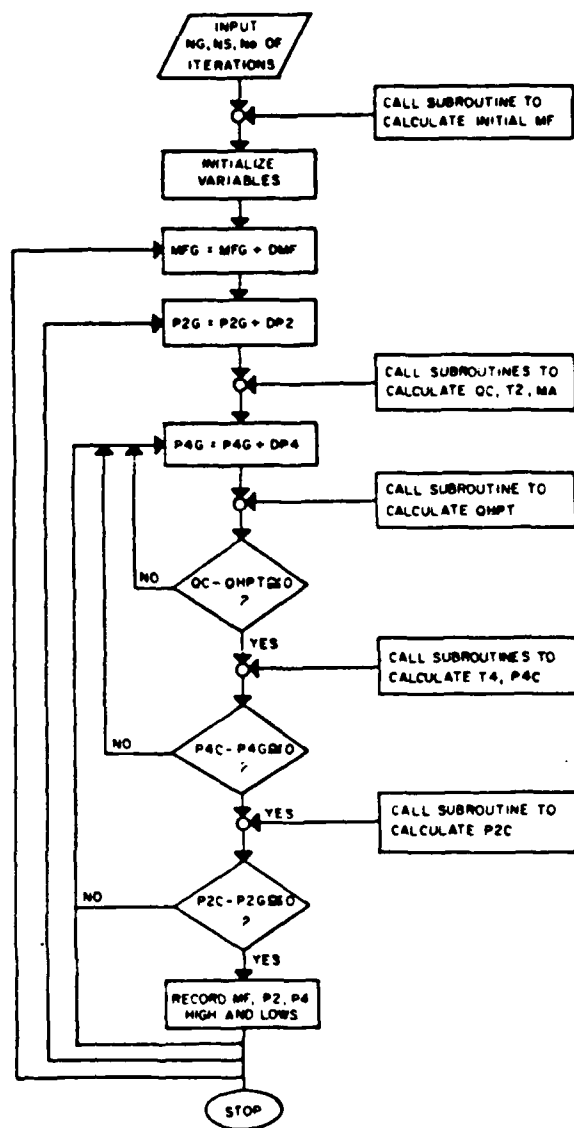


Figure 7. Flowchart of VRD Algorithm

values of the various increments as well as to initialize the high and low values. The VRD increments were established using the maximum and minimum possible values in the normal plant operating envelope for each of the three variables being considered. The first fuel guess was arbitrarily set 20 pounds less than the value returned by NGNSMF to ensure proper coverage of the beginning of the fuel range. The fuel iteration loop then incremented the fuel and set P2 to the low value of its operating range. The P2 loop was then entered, P2G incremented, and subroutines called to compute compressor torque (QC), compressor discharge temperature (T2), and air mass flow rate (MA). The P4 low value was then set, P4 loop entered, and P4G incremented. The high pressure turbine torque (QHPT) was calculated via a subroutine, and then QC and QHPT were compared. If the difference had not changed sign, torque convergence had not occurred and P4G was incremented. If the difference had changed sign or was equal to zero, torque convergence was assumed and subroutines to calculate high pressure turbine discharge temperature (T4) and P4C were called. P4G and P4C were then compared for convergence in a similar fashion. If convergence was not obtained, P4G was incremented and the process continued. If P4 convergence was obtained the P2 subroutine was called upon to compute P2C, then the third and last convergence check was made on P2. Failure to converge incremented P4G.

Successful convergence in all three tests resulted in logic which identified the variable range for possible solution(s). Upon completion of the P4 range, the P2 loop was incremented and the process repeated. The MF loop was incremented when the range of P2 values was exhausted, and the routine continued until the MF range had been traversed. The end result was that for every incremented value of MF, all combinations of P2G and P4G were examined and compared to computed values. The results were then grouped to provide the solution variable ranges for the second algorithm, which refined the solution.

The Solution Convergence (SC) algorithm was essentially the same as the previous VRD algorithm. A copy of the SC program is included as Appendix B. The high and low values for MF, P2, and P4 were specified to be those obtained from the VRD algorithm. The increments in the SC algorithm were defined as functions of the VRD high and low variables and the originally defined VRD increments. The SC increments were smaller than the VRD increments, providing a finer grid to be searched. The search range for each variable was established by starting one VRD increment outside the initial value and then incrementing by SC increments. This method ensured proper coverage in the vicinity of the initial value of the range. Coverage was extended slightly past the final value by adding an arbitrary number of iterations to the number of "guesses" specified for each

variable during the input phase. Convergence logic was the same as previously discussed. The accuracy of the converged solution was determined by the magnitude of the terms DELQ (QC - QHPT), DELP2 (P2C - P2G), and DELP4 (P4C - P4G). The smaller the "DEL" terms, the more accurate the solution. The output of the routine was a list of all converged solutions that lay within the specified ranges of MF, P2, and P4.

With the critical convergence criteria met, the final portion of the steady state solution process could be completed. A third computer routine was developed from work performed by Herda and Miller. The Steady State Variable and Partial Derivative (SSVPD) Algorithm used the converged MF, P2, and P4 values to compute steady state values for air mass flow (MA), air-fuel mass flow (MAF), high pressure turbine discharge temperature (T2), power turbine discharge temperature (T4), free power turbine torque (QFPT), dynamometer torque (QD), high pressure turbine torque (QHPT), compressor torque (QC), and dynamometer water weight (WW). SSVPD calculated the partial derivatives required for the necessary linearizations to form the state space "A" and "B" matrices. A copy of the SSVPD program is included as Appendix C.

Using this three step process, it was possible to converge steady state solutions for all gas generator speeds between 22000 RPM and 32000 RPM, and for dynamometer speeds

between 500 RPM and 2500 RPM. Gas generator speeds below 22000 RPM were sporadically convergeable, they were unconvergeable below 20000 RPM and above 35000 RPM. Power turbine speeds above 2500 RPM were not consistently convergeable.

Miller hypothesized that quadratic data fits to data for the various components of the model may not be valid throughout the operating envelope, and this work seems to validate that theory. That is, quadratic fits appear to be reasonable in the middle of the operating envelope, but not in the low or high portions.

A Steady State Convergence Map of "A" matrices was constructed for the convergeable region of the operating envelope, and is shown in Figure 8. Since the states NG and NS characterize the plant in state space, these variables were chosen as the coordinate axes of the grid. For each node of the grid, the list of converged solutions from the SC routine was examined and the DELQ, DELP2, and DELP4 values compared. Convergence accurate to 0.1 pound of fuel and 0.1 psi for both pressures were set as minimums or the solution was eliminated from the list. All remaining candidates for each node were subsequently entered into the SSVPD algorithm and the results collected. Strip chart recordings of actual plant runs were examined and those runs lying in the convergeable region were utilized. The start and end points of these runs were converged and used to

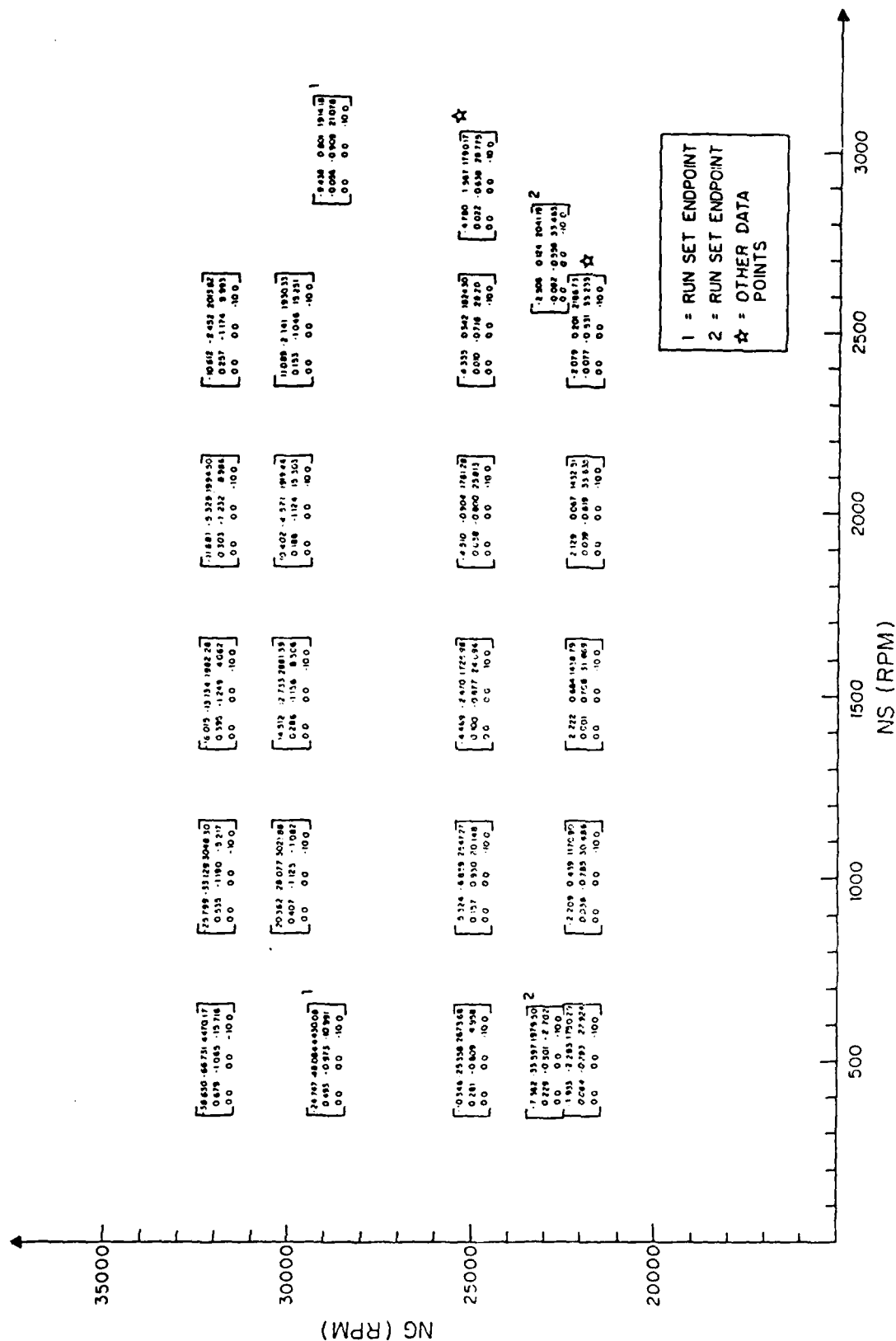


Figure 8. Steady State Convergence Map

anchor the grid. Once anchored, the remaining grid points were selected by comparing the various matrix coefficients for trends both horizontally along lines of constant NG and vertically along lines of constant NS. The sensitivity to convergence accuracies was demonstrated by the wide variance in magnitude of the matrix coefficients for a given grid point, particularly the A13 and A23 entries. The A13 entry was extremely sensitive to P4. By establishing the horizontal and vertical trends on the grid, it was a relatively simple matter to select/adjust a particular grid point from the various candidates until the best overall result was obtained.

V. DYNAMIC SIMULATION

Due to very low gas turbine inertias, the smoothness of the changes in "A" matrix entries was critical to effective plant simulation. Consequently, the Steady State Convergence Map was first analyzed for horizontal and vertical trends, both in magnitude and sign. Overall trends were readily apparent with the exception of seven A12 entries, five A21 entries, and five A23 entries. Of these discrepancies one was an ill fitting data point (the A23 entry of the matrix at NG = 23150 rpm and NS = 493 rpm was of the wrong magnitude) and consequently the entire matrix was disregarded, while the remainder were of the correct magnitude but of the wrong sign. Table 1 is a summary of the matrix coefficients that were of the wrong sign. Possible reasons for these errors include poor data fit at low engine and dynamometer speeds, and the unreliability of data values recorded at low engine horsepowers as documented in the dynamometer technical manual (13). Also, these values were observed to be extremely sensitive to P4 which had a large convergence tolerance.

Further examination of the Steady State Convergence Map revealed that relatively smooth curves could be generated both horizontally and vertically along lines of constant NG and NS for each matrix coefficient. The Smoothed Dynamic

TABLE 1
STEADY STATE CONVERGENCE MAP "A" MATRIX
COEFFICIENT SIGN ERRORS

GRID LOCATION (NG,NS)	MATRIX COEFFICIENT	CORRECT SIGN
22000,1000	A12	-
22000,1500	A12	-
22000,2000	A12	-
22000,2000	A21	+
22000,2500	A21	+
23150,2695	A12	-
23150,2695	A21	+
25000,2500	A12	-
25000,3000	A12	-
25000,3000	A21	+
29100,500	A23	+
29100,2950	A12	-
29100,2950	A21	+
3000,1000	A23	+
32000,500	A23	+
32000,1000	A23	+

Transition Map of Figure 9 was formed by "eyeball smoothing" the data points from the Steady State Convergence Map. Trends were easily seen in the middle and right side of the Steady State Convergence Map, and these were used as models in the areas where sign discrepancies existed. The end

result was a grid of smoothed "A" matrices that would be the cornerstone of the dynamic simulation.

A two-variable linear regression computer algorithm was used to obtain the coefficients of the best curve fit for each corresponding set of matrix coefficients. Table 2 is a summary of the six fits used. The A31, A32, and A33 entries of the "A" matrices were those that form the state variable corresponding to combustion energy and were always the same, hence no fit was required. The "B" matrix was constant at all values of NG and NS.

The dynamic simulation of the plant was conducted using an IBM mainframe computer routine entitled Dynamic Simulation Language (DSL). The above described "A" matrix validation led to a method of successive linearizations to compute the values of NG and NS at time intervals of 0.001 seconds during the dynamic trajectories being modelled.

Strip chart data from actual plant runs was entered into the DSL code to provide the curve for model comparison. The initial and final plant setpoints were then specified, followed by the equations to compute the various matrix coefficients, the linearization equations, and the various output and graphing statements required. Figure 10 depicts single and multiple linearizations plotted against data for a dynamometer speed versus time curve. Clearly, multiple linearizations were necessary to ensure the proper ending steady state values were reached. Appendix E is a copy of

TABLE 2
"A" MATRIX COEFFICIENT CURVE FITS

MATRIX COEFFICIENT	CURVE FIT	EQUATION FORM
A11	Exponential	$A11 = \text{EXP}(C1*NS+C2*NG+C3)$
A12	Exponential	$A12 = \text{EXP}(C1*NS+C2*NG+C3)$
A13	Exponential	$A13 = \text{EXP}(C1*NS+C2*NG+C3)$
A21	Exponential	$A21 = C1*NS^2+C2*NS*NG$ $+C3*NG^2+C4*NS$ $+C5*NG+C6$
A22	Linear	$A22 = C1*NS+C2*NG+C3$
A23	Exponential	$A23 = \text{EXP}(C1*NS)+C2*NG+C3)$

the dynamic simulation program. Appendix F compares the "A" matrix coefficients of the Smoothed Dynamic Transition Map to those obtained by the dynamic simulation.

Model validation was conducted in the region of the Smoothed Dynamic Transition Map known to be most reliable. Three runs were made across the map at constant NG speeds of 23150, 29100, and 34900 rpm with varying NS speeds. The results are shown in Appendix G. All show excellent agreement between the model and data. A fourth validation was attempted vertically on the left side of the smoothed grid, starting at NG = 17000.0 rpm and NS = 500 rpm and ending at NG = 35500 rpm and NS = 748 rpm. The results obtained were less than satisfactory. However, this particular validation effort began and ended well outside of

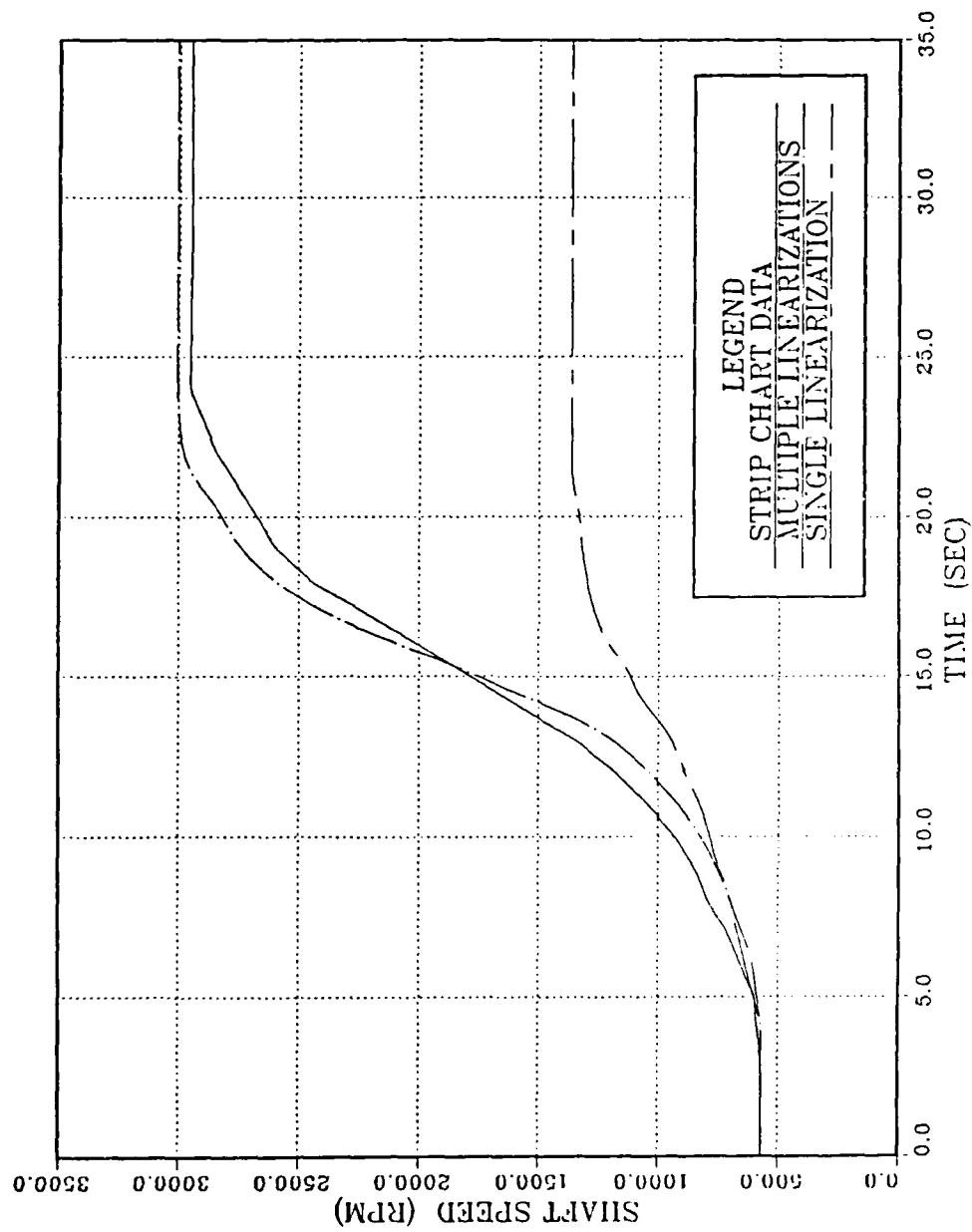


Figure 10. Effect of Linearizations on NS

the region considered to be reliable on the "A" map, and progressed through the unreliable low dynamometer speed range. For these reasons the speed ranges chosen for the controller design and validation portion of this work were in the center of the Smoothed Dynamic Transition Map which was considered validated.

VI. CONTROLLER DESIGN AND COMPARISON

The final segment of this work was concerned with controller design and comparison for overall performance and disturbance rejection qualities. Two controllers were designed, one a classical proportional integral controller (PI) and the other a modern linear quadratic regulator (LQR). Due largely to the fact that marine gas turbine controller designs are largely proprietary in nature, the design procedures used in this work had to be developed by the author.

Deceleration was chosen as the design case because initial work clearly showed a much smaller stability margin than acceleration operations. Each controller was designed and validated for varying gas generator and shaft speed, but at constant dynamometer water volume. These specifications simulate acceleration and deceleration modes at constant propeller pitch. Deceleration began at steady state speeds of NG = 34900 rpm and NS = 2000 rpm, and were subsequently slowed to NG = 23150 rpm and NS = 1500 rpm. The values were simply reversed for the acceleration case.

The control scheme shown in Figure 1 is currently employed by the U.S. Navy for control of General Electric LM-2500 gas turbine propulsion plants aboard DD-963 class destroyers. Note that the Navy uses a two loop approach to

control the gas turbine speed, propulsion shaft speed, and propeller pitch. Inputs to the plant were through an integrated throttle arrangement that schedules these three quantities for the desired ship speed. In order to demonstrate a similar control concept, the plant model used in the present work was chosen to closely emulate the general structure of Figure 1. Figure 11 is a diagram of this model and the control structure used for the PI design.

The PI controller was designed first using a two step process. The inner loop consisted of a proportional control scheme to govern the gas generator. It was designed prior to the outer loop, using a cut and try process to select the appropriate gain (KPNG) to give a smooth non-oscillatory response. Steady state error was not a major consideration in the inner loop because the overall plant control was to be exerted on the propulsion shaft. Figures 12 and 13 depict different responses for various choices of KPNG for deceleration and acceleration respectively. A gain of $KPNG = 0.001$ was chosen for the inner loop and held constant throughout the design of the outer loop.

A proportional integral arrangement was employed for the outer NS control loop. This was chosen so that the steady state error associated with shaft response would be eliminated. Once again a cut and try procedure was used to decide the shaft proportional gain (KPNS) and integral (KINS) gain, with the criteria of smooth response driving

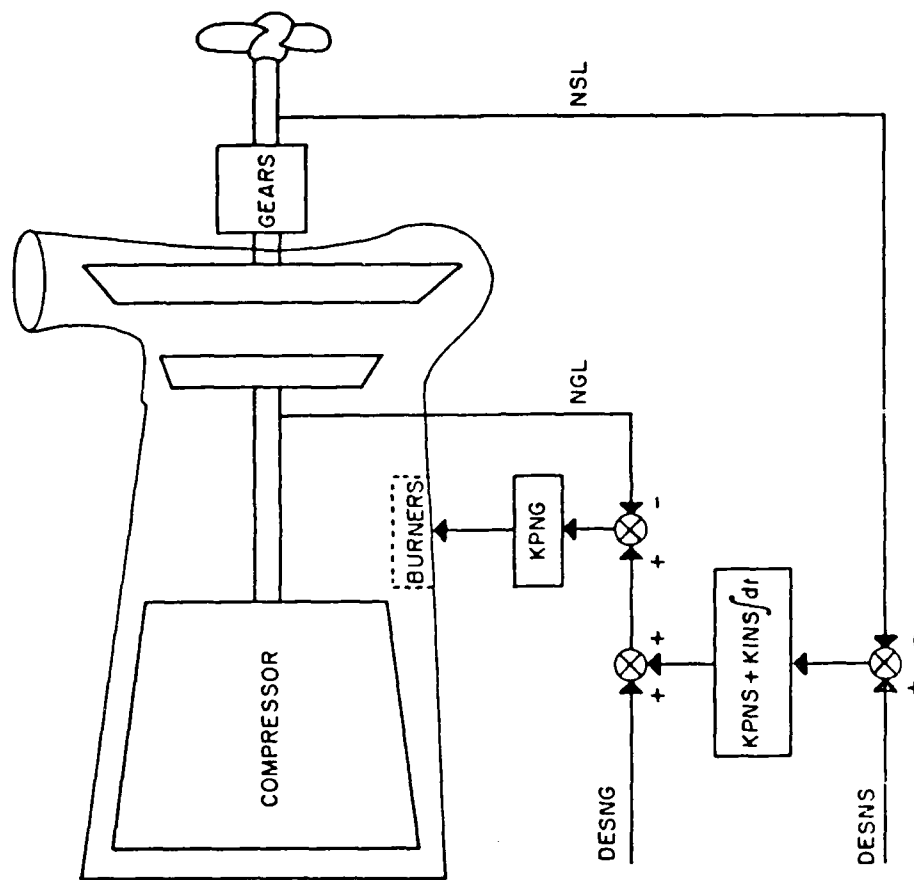


Figure 11. Author's PI Controller--Plant Configuration

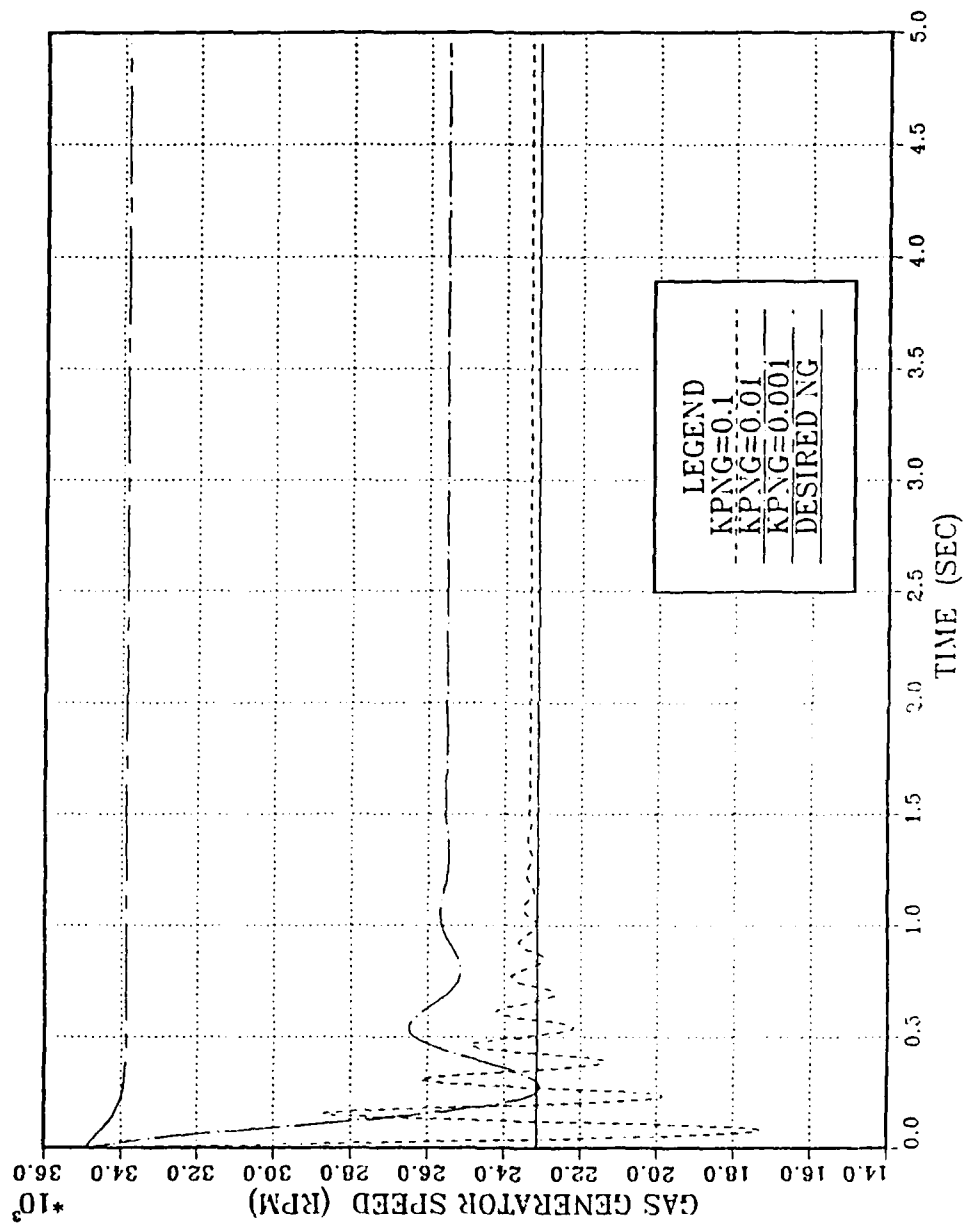


Figure 12. Proportional Control--Deceleration

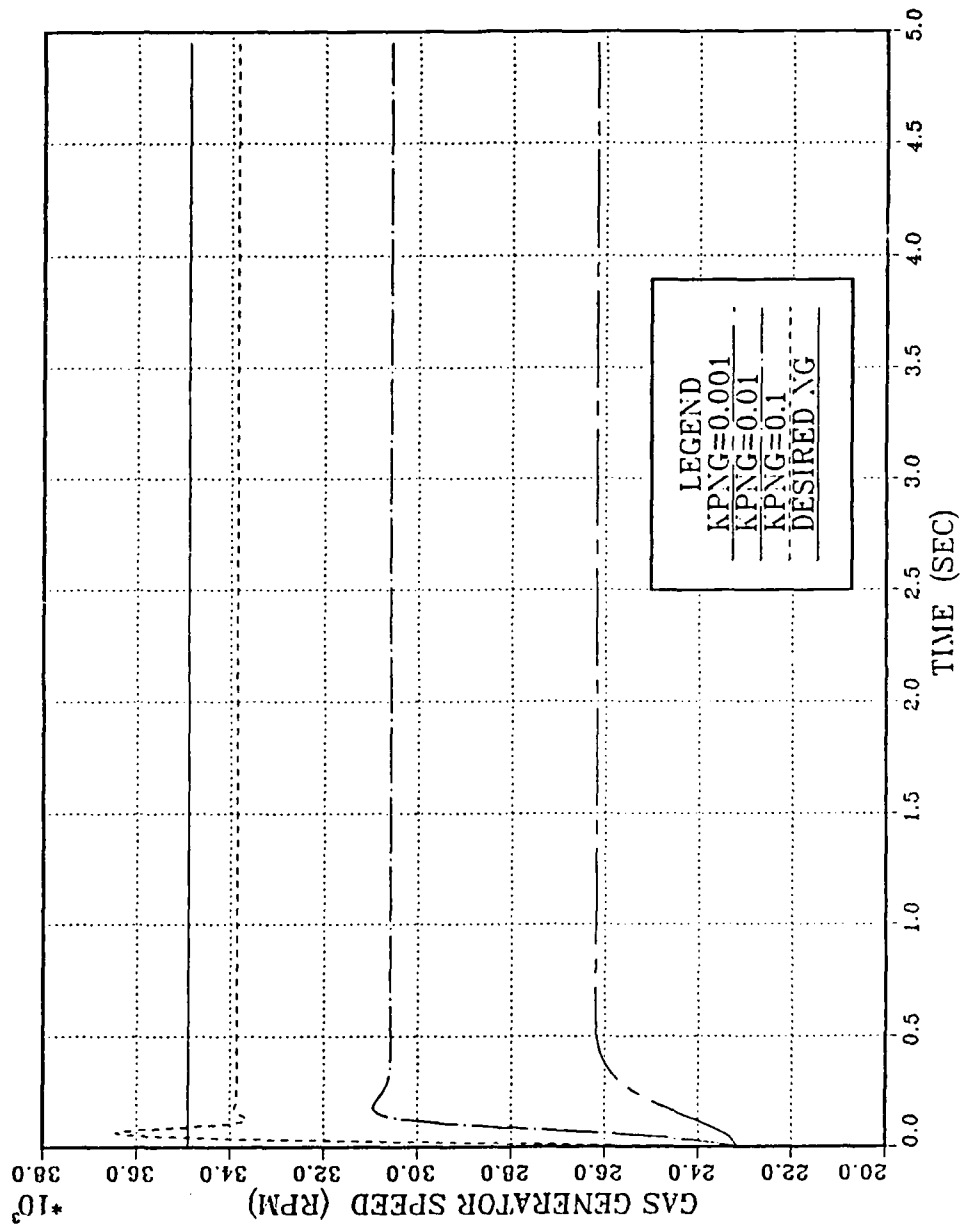


Figure 13. Proportional Control--Acceleration

the selections. Figures 14 and 15 show plant response to various gains for the deceleration case in the gas generator and power turbine output shaft respectively. Gains of KPNS = 80.0 and KINS = 40.0 were selected for the outer control loop.

The LQR controller was designed next using the control design software package MATRIXX. State space "A" and "B" matrices at the center of the smooth grid (NG = 25000 rpm, NS = 1500 rpm) were chosen to fix the design. In the LQR method, gains are sought to minimize a specified performance index "J" (or cost function) [Ref. 14]. The performance index is expressed as an integral containing a function of the state variables and a function of the input variables:

$$J = \int_0^{\infty} (\tilde{e}^T \tilde{Q} \tilde{e} + \tilde{u}^T \tilde{R} \tilde{u}) dt \quad (3)$$

The "Q" and "R" matrices are symmetric weighting matrices that weight the states and inputs respectively. The designer chooses "Q" and "R", then computes the performance index which results in the LQR gains. Tradeoffs between "Q" and "R" weightings can be performed to achieve the best results.

In the present work, the "Q" matrix was scaled so that each state matrix was making an approximately equal contribution to the response. The "R" matrix was chosen by a cut and try process, and was designed to minimize the

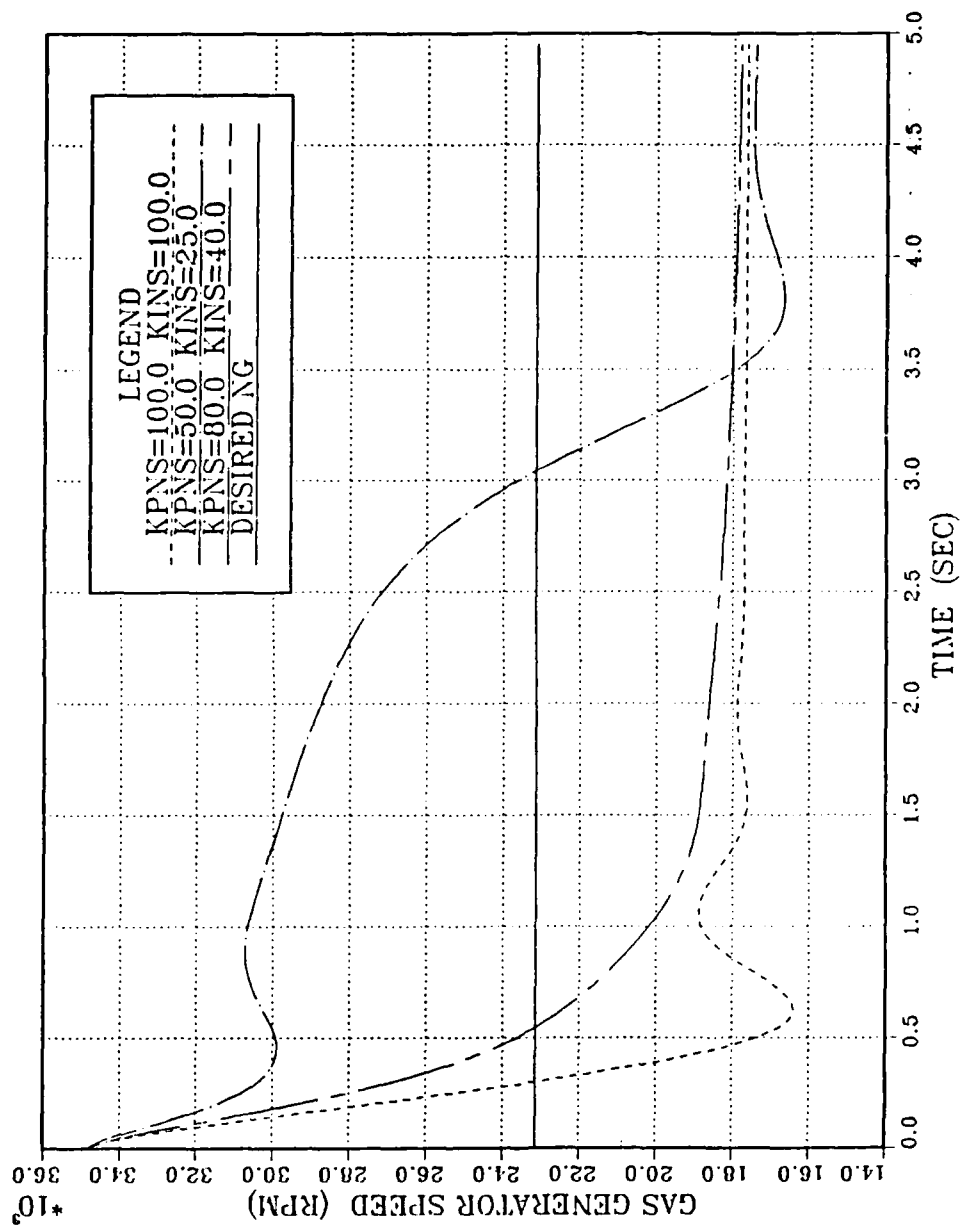


Figure 14. Proportional Integral Control--NG, Deceleration

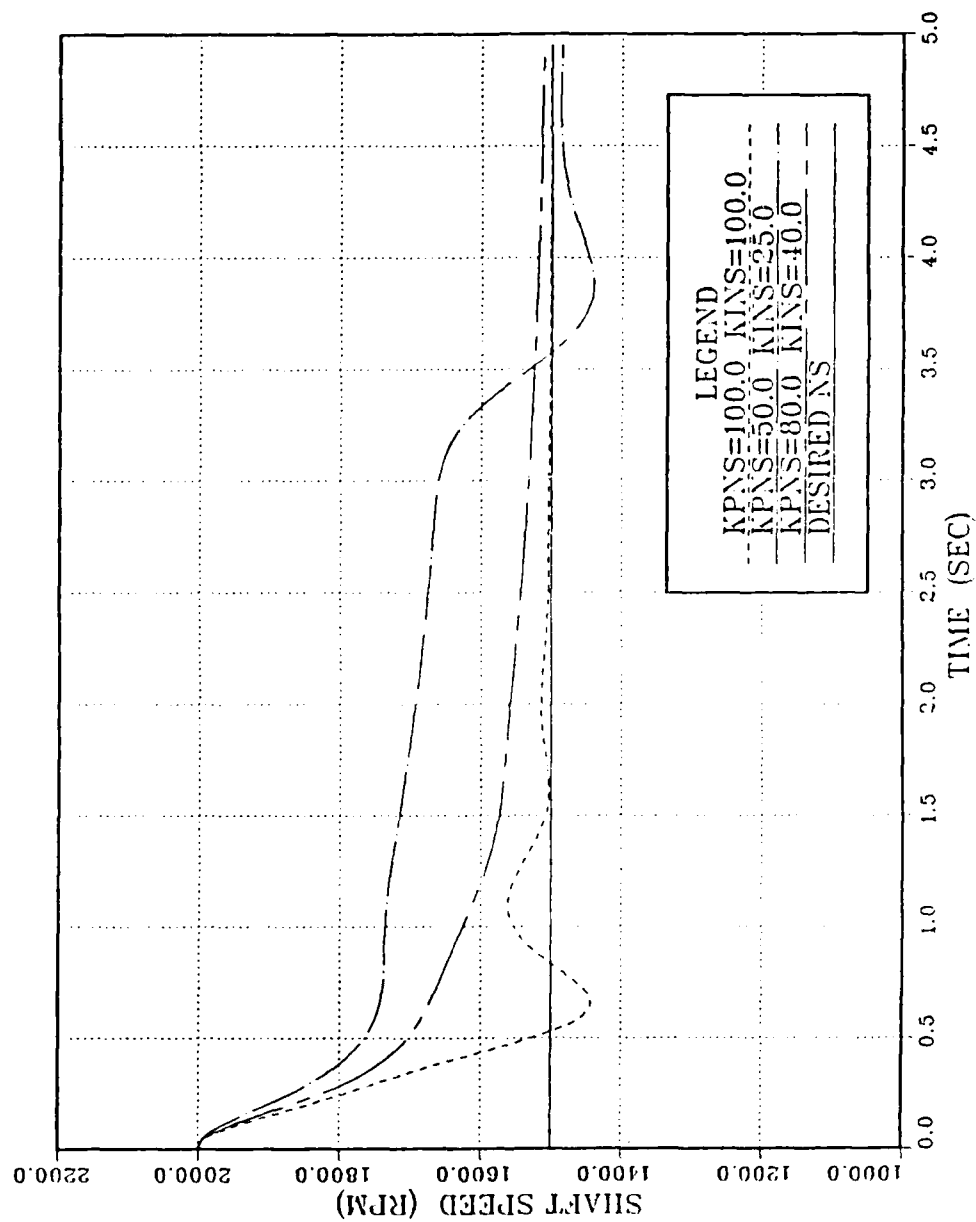


Figure 15. Proportional Integral Control--NS, Deceleration

plant fuel input. The matrices were adjusted until the most acceptable response was obtained. This particular LQR design is strictly a proportional regulator and has no integral action to remove steady state error. As a result there is some steady state error in both NG and NS. It was possible to eliminate or nearly eliminate this error in either NG or NS, but at the expense of the other variable. The chosen design exhibits small steady state errors in both NG and NS.

Figures 16, 17, 18, and 19 are comparisons of the deceleration and acceleration validations of the PI and LQR controllers for both the gas generator and power turbine shafts. The PI controller provides a smoother response in both NG and NS deceleration curves, while LQR reaches steady state in less time. The acceleration validation shows the LQR controller providing a smoother, quicker response in NS and a quicker response in NG. In actuality, all responses are comparable for both controllers.

Disturbance rejection for both controllers was analyzed by subjecting each one to a cyclic torque disturbance simulating sea state oscillations. A sine function load was used that provided a ten second period:

$$Q_L = 20.0 \sin(t\pi/5.0) \quad (4)$$

The controllers were set to maintain steady gas generator and shaft speeds of NG = 17642 rpm and NS = 1500 rpm

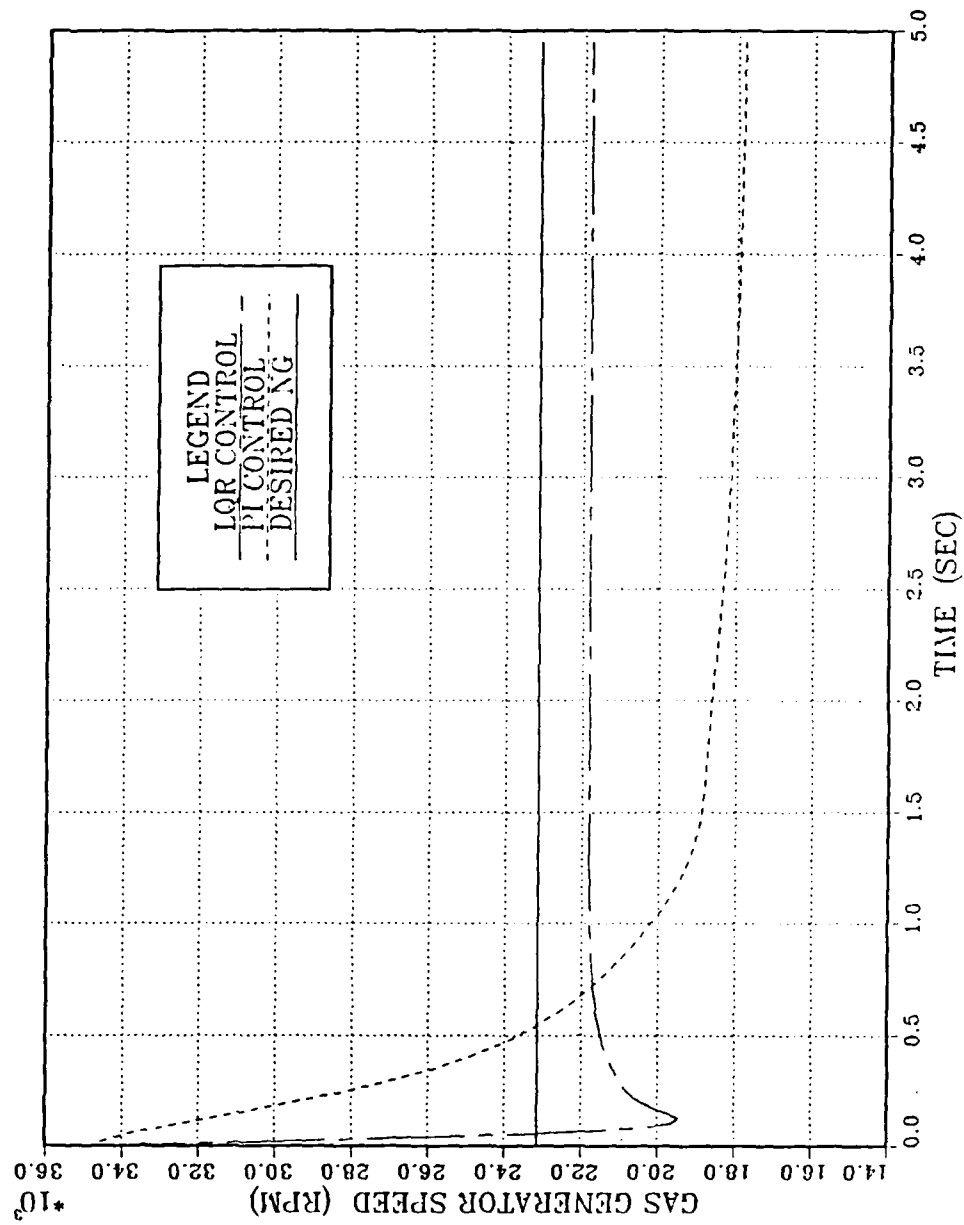


Figure 16. Controller Comparison--NG, Deceleration

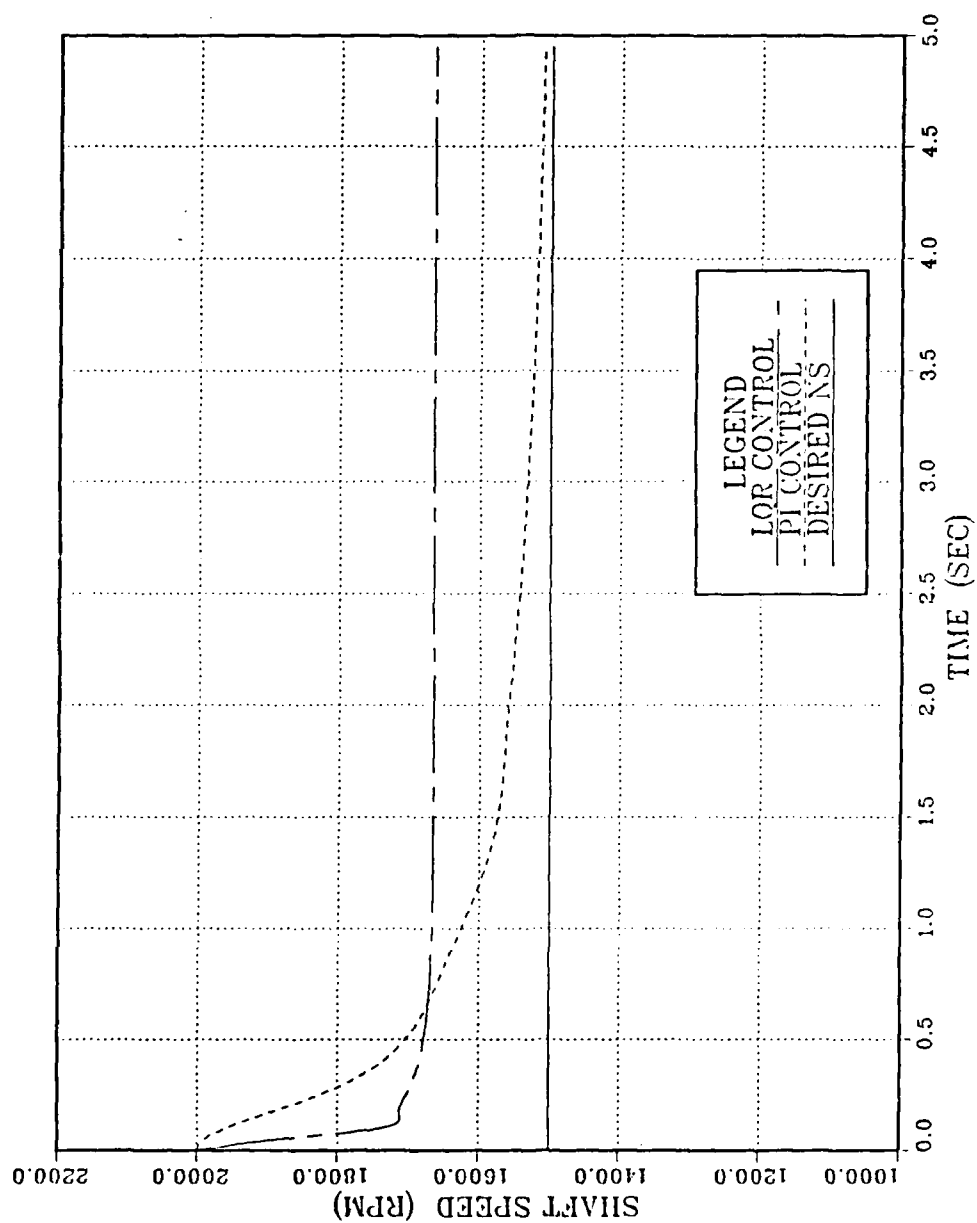


Figure 17. Controller Comparison--NS, Deceleration

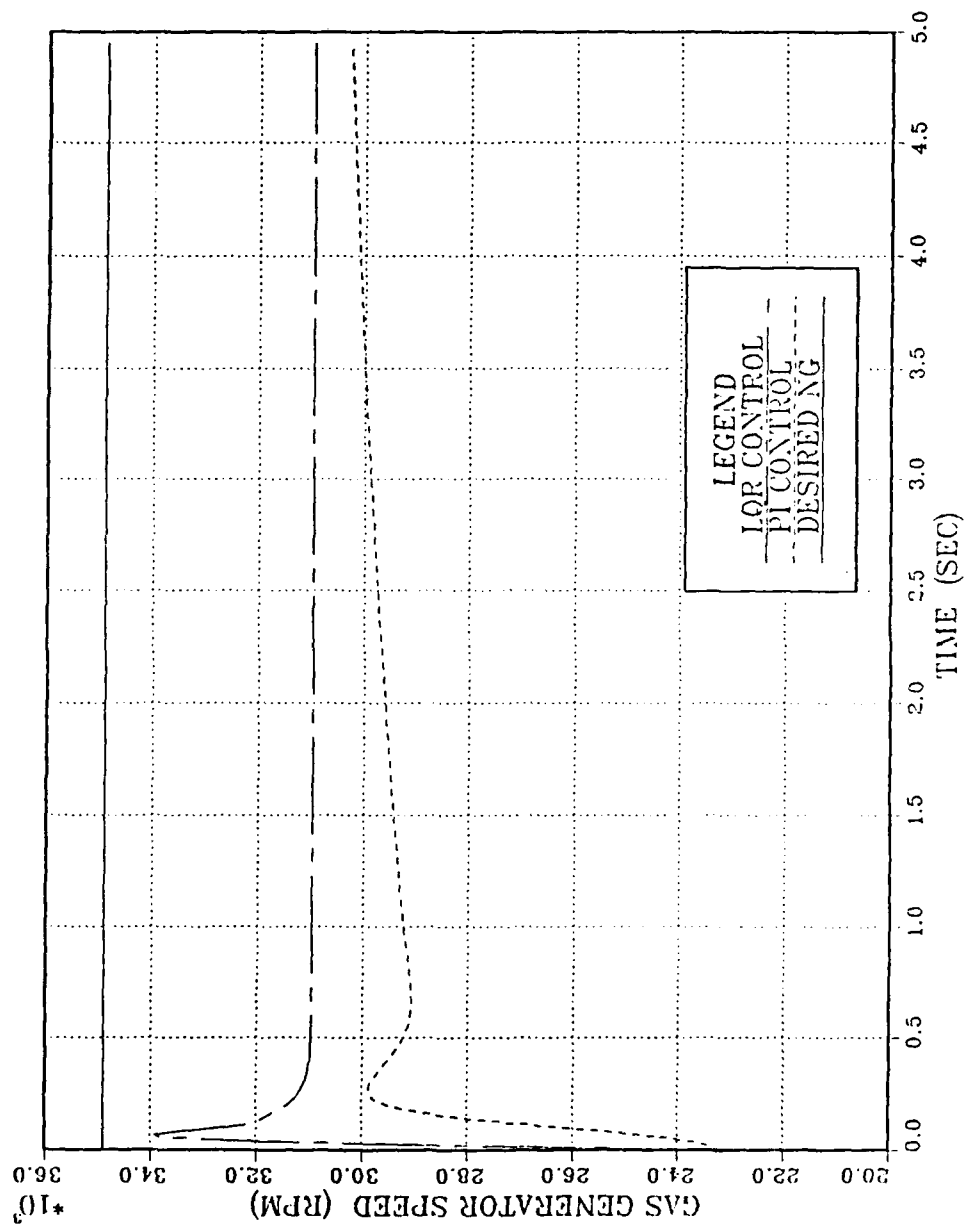


Figure 18. Controller Comparison--NG, Acceleration

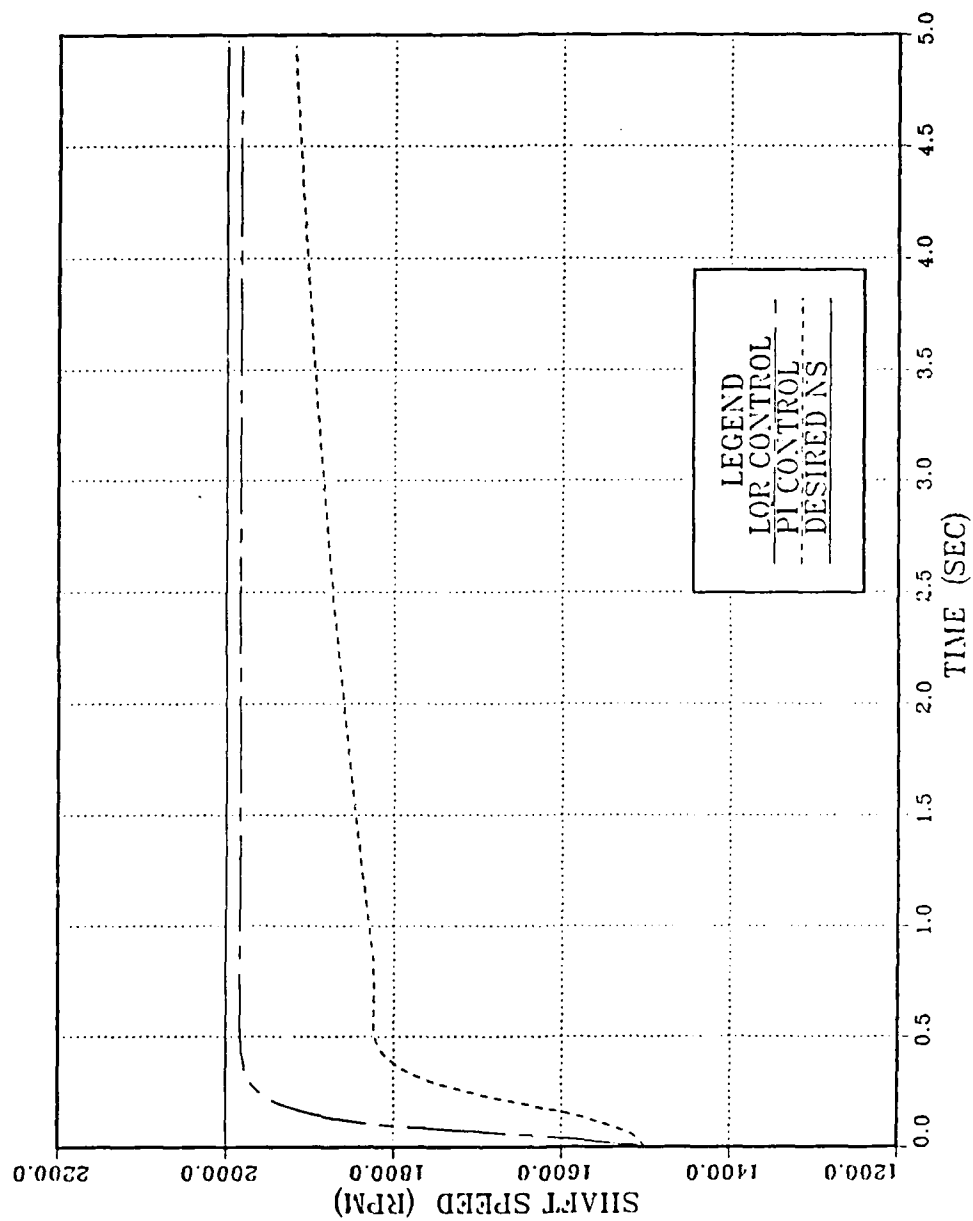


Figure 19. Controller Comparison--NS, Acceleration

respectively, and the Q_L function applied through the torque input to the models. Each model was run for 35 seconds and the data recorded. Peak-to-peak values of NG and NS were obtained, as well as fuel consumed by the plant for both control designs. Table 3 compares the results for two LQR designs to the PI design and graphically illustrates the effects that the before mentioned tradeoffs can have on the LQR design. Of particular interest are the NG peak-to-peak values and the fuel consumed values. The smaller NG peak-to-peak values indicate better disturbance rejection for the NG output, which translates to less wear on critical gas turbine components such as variable stator vanes. This in turn has the potential to decrease maintenance downtime as well as mean time between failures of complex mechanical components. Although the difference in fuel consumed may be small, the potential exists for substantial fleetwide reductions in fuel costs when this figure is applied to the amount of fuel burned annually by all gas turbine ships.

TABLE 3
CONTROLLER DESIGN COMPARISON

	NG ¹	NS ¹	FUEL ²
LQR #1	3320.0	203.1	0.74136
LQR #2	3688.0	125.8	0.74208
PI	4233.0	74.5	0.74337

¹Peak-to-Peak Values, rpm

²Total lb_m consumed in 35 seconds

VII. CONCLUSIONS AND RECOMMENDATIONS

Two non-proprietary control design methods have been developed for marine gas turbine propulsion systems, one a classical PI controller, the other a modern LQR controller.

Comparable performance has been demonstrated between the PI and LQR controllers.

A new gas turbine simulation has been developed that allows real time or near real time computation of performance. This simulation has immediate applications in model based control and plant health monitoring.

To increase confidence in the plant model, the Steady State Convergence Map should be reconstructed from data obtained from actual plant runs at designated points throughout the operating envelope. This would eliminate the problems of multiple root convergence in the steady state computer simulations, and provide a more accurate data base for the curve fits required by the dynamic simulation.

Proportional LQR design should be further developed to account for important non linearities in the gas generator and controllable reversible pitch propeller, as well as limiting/alarm conditions that exist in fleet marine gas turbine propulsion plants.

An integral LQR controller should be investigated. This type of controller has the potential to offer better

performance tradeoffs in the areas of gas generator speed control, shaft speed control, and in the reduction of fuel consumption.

Once the LQR controller is perfected, the idea of an LQR Gain Map similar to the Smoothed Dynamic Transition Map should be investigated. This concept has the potential to maximize the benefits of the LQR controller by computing the best gain values at each time step as the plant progresses through a particular dynamic trajectory.

This technology should be implemented at the Naval Postgraduate School to quantify real improvements and justify larger scale development.

APPENDIX A

```

*****
*                               *
*           ROUTINE  VRD       *
*                               *
*                               *
*           BY                 *
*                               *
*           V. A.  STAMMETTI   *
*                               *
*           D. L.  SMITH       *
*                               *
*                               *
*           THIS ROUTINE PROVIDES THE FIRST OF A THREE STEP
*           PROCESS TO OBTAIN STEADY STATE CONVERGENCE AT A POINT
*           IN THE OPERATING ENVELOPE OF THE BOEING 501-6A GAS
*           TURBINE INSTALLED AT THE NAVAL POSTGRADUATE SCHOOL.
*           ROUTINE VRD IDENTIFIES THE SEARCH RANGES FOR THE
*           VARIABLES MF, P2, AND P4.
*                               *
*                               *
*****
C
C
C           REAL  NG,NS,MF,MA,MAF,MFG,MFI,MFHIGH,MFLOW,MFSAVG
C
C           INPUT THE INITIAL GAS GENERATOR SPEED AND DYNO SPEED.
C
C           WRITE(6,2)

```



```

2  FORMAT(/,3X,'INPUT INITIAL GAS GENERATOR SPEED,"NG".')
C
    READ(5,*) NG
    WRITE(6,*) NG
C
C
    WRITE(6,3)
3  FORMAT(/,3X,'INPUT INITIAL DYNO SPEED,"NS".')
C
    READ(5,*) NS
    WRITE(6,*) NS
C
    WRITE(9,*) '  NG=',NG
    WRITE(9,*) '  NS=',NS
C
C
    WRITE(6,4) NMF
4  FORMAT(/,3X,'INPUT NUMBER OF FUEL GUESSES,"NMF".')
    READ(5,*) NMF
    WRITE(6,*) NMF
C
C
    WRITE(6,5) NP2
5  FORMAT(/,3X,'INPUT NUMBER OF P2 GUESSES,"NP2".')
    READ(5,*) NP2
    WRITE(6,*) NP2
C

```

C

WRITE(6,6) NP4

6 FORMAT(/,3X,'INPUT NUMBER OF P4 GUESSES,"NP4".')

READ(5,*) NP4

WRITE(6,*) NP4

C

C

CALCULATION OF INITIAL FUEL GUESS

CALL NGNSMF(NG,NS,MFI)

C

C

VARIABLE INITIALIZATION

C

MFG = MFI - 20.0

RNMF = NMF

RNP2 = NP2

RNP4 = NP4

DMF = 40.0 / RNMF

DP2 = 25.0 / RNP2

DP4 = 5.0 / RNP4

QCONV = 10.0

P2CONV = 0.5

P4CONV = 0.5

MFLOW = MFI

MFHIGH = MFG

P4LOW = 100.0

P4HIGH = 1.00

P2LOW = 100.0

P2HIGH = 1.00

ATEST = 1000.0

```

P2SAVC = 0.0
P2SAVG = 0.0
P4SAVC = 0.0
P4SAVG = 0.0
MFSAVG = 0.0

C
C   MF LOOP
C

DO 100 J = 1,NMF
MFG = MFG + DMF
P2SAVE = 0.0
P2G = 20.0
WRITE(6,*)
WRITE(6,*) 'MFG=',MFG

C
C   P2 LOOP
C

DO 200 K = 1,NP2
P2G = P2G + DP2
CALL SUBQC(NG,P2G,QC)
CALL SUBT2(NG,P2G,T2)
CALL SUBMA(NG,P2G,MA)
QSAVE = 0.0
P4SAVE = 0.0
P4G = 20.0

C
C   P4 LOOP
C

```

```

DO 300 L = 1,NP4
P4G = P4G - DP4
C
C TORQUE CONVERGENCE
C
CALL SUBQHT(NG,MA,T2,MFG,P4G,QHPT)
DELQ = QC - QHPT
QTEST = DELQ * QSAVE
QSAVE = DELQ
IF(QTEST.GE.0.0) GO TO 300
C
C P4 CONVERGENCE
C
10 MAF = MA + MF
CALL SUBT4(NG,MA,T2,MFG,P4G,T4)
CALL SUBP4(MAF,T4,NS,P4C)
DELP4 = P4C - P4G
P4TEST = DELP4*P4SAVE
P4SAVE = DELP4
IF(P4TEST.GE.0.0) GO TO 300
C 15 IF(ABS(DELP4).GT.P4CONV) GO TO 300
C
C P2 CONVERGENCE
C
20 CALL SUBP2(NG,MA,T2,MFG,P4G,P2C)
DELP2 = P2C - P2G
P2TEST = DELP2*P2SAVE
P2SAVE = DELP2

```

IF(P2TEST.GE.0.0) GO TO 300

C

30 IF(MFG.LT.MFLOW) MFLOW = MFG
IF(MFG.GT.MFHIGH) MFHIGH = MFG
IF(P4G.LT.P4LOW) P4LOW = P4G
IF(P4G.GT.P4HIGH) P4HIGH = P4G
IF(P2G.LT.P2LOW) P2LOW = P2G
IF(P2G.GT.P2HIGH) P2HIGH = P2G

C

DELSUM = (DELQ**2) + (DELP2**2) + (DELP4**2)
IF(DELSUM.GE.ATEST) GO TO 300

35 P2SAVC = P2C
P2SAVG = P2G
P4SAVC = P4C
P4SAVG = P4G
MFSAVG = MFG
DELQG = DELQ
DELP2G = DELP2
DELP4G = DELP4
ATEST = DELSUM

C

WRITE(6,*) ' CONVERGENCE OBTAINED AT'
WRITE(6,*) ' P2C=',P2SAVC
WRITE(6,*) ' P2G=',P2SAVG
WRITE(6,*) ' P4C=',P4SAVC
WRITE(6,*) ' P4G=',P4SAVG
WRITE(6,*) ' MF=',MFSAVG
WRITE(6,*) ' DELQ=',DELQG

```

WRITE(6,*) ' DELP2=',DELP2G
WRITE(6,*) ' DELP4=',DELP4G

C
C
C
300    CONTINUE
C
C
200    CONTINUE
100    CONTINUE
C

WRITE(6,*) ' CONVERGENCE OBTAINED AT'
WRITE(6,*) ' P2C=',P2SAVC
WRITE(6,*) ' P2G=',P2SAVG
WRITE(6,*) ' P4C=',P4SAVC
WRITE(6,*) ' P4G=',P4SAVG
WRITE(6,*) ' MF=',MFSAVG
WRITE(6,*) ' DELQ=',DELQG
WRITE(6,*) ' DELP2=',DELP2G
WRITE(6,*) ' DELP4=',DELP4G

C

WRITE(6,*) ' MFLOW=',MFLOW
WRITE(6,*) ' MFHIGH=',MFHIGH
WRITE(6,*) ' P4LOW=',P4LOW
WRITE(6,*) ' P4HIGH=',P4HIGH
WRITE(6,*) ' P2LOW=',P2LOW
WRITE(6,*) ' P2HIGH=',P2HIGH

```

C

```
WRITE(9,*) ' CONVERGENCE OBTAINED AT'  
WRITE(9,*) ' P2C=',P2SAVC  
WRITE(9,*) ' P2G=',P2SAVG  
WRITE(9,*) ' P4C=',P4SAVC  
WRITE(9,*) ' P4G=',P4SAVG  
WRITE(9,*) ' MF=',MFSAVG  
WRITE(9,*) ' DELQ=',DELQG  
WRITE(9,*) ' DELP2=',DELP2G  
WRITE(9,*) ' DELP4=',DELP4G
```

C

```
WRITE(9,*) ' MFLOW=',MFLOW  
WRITE(9,*) ' MFHIGH=',MFHIGH  
WRITE(9,*) ' P4LOW=',P4LOW  
WRITE(9,*) ' P4HIGH=',P4HIGH  
WRITE(9,*) ' P2LOW=',P2LOW  
WRITE(9,*) ' P2HIGH=',P2HIGH
```

C

```
900 STOP  
END
```

APPENDIX B

```

*****
*                               ROUTINE  SC                               *
*                               BY                                           *
*                               V. A. STAMMETTI                             *
*                               D. L. SMITH                                 *
*                               *                                           *
*   THIS ROUTINE PROVIDES THE SECOND OF A THREE STEP                     *
*   PROCESS TO OBTAIN STEADY STATE CONVERGENCE AT A POINT                 *
*   IN THE OPERATING ENVELOPE OF THE BOEING 501-6A GAS                   *
*   TURBINE INSTALLED AT THE NAVAL POSTGRADUATE SCHOOL.                 *
*   ROUTINE SC CONVERGES THE VARIABLES MF, P2, AND P4.                   *
*                               *                                           *
*****
C
C
      REAL*8  NG,NS,MF,MA,MAF,MFG,MFI,MFSAVG,MFHIGH,MFLOW,MFL,RNMF,
1  RNP2,QCONV,P2CONV,P4CONV,P2LOW,P2HIGH,P4LOW,P4HIGH,ATEST,
1  P2SAVE,P2SAVC,P2SAVG,P4SAVE,P4SAVC,P4SAVG,DMF,DP2,DP4,RNNMF,
1  DMF2,P2L,RNNP2,DP22,P4H,RNNP4,DP42,P2G,QC,T2,QSAVE,P4G,P4C,
1  P2C,QHPT,DFLQ,QTEST,T4,DELP4,P4TEST,P2TEST,DELP2,DELSUM,
1  RNP4,DELQG,DELP4G,DELP2G
C
C
      INPUT THE INITIAL GAS GENERATOR SPEED AND DYNO SPEED.
C
C
      WRITE(6,2)
2  FORMAT(/,3X,'INPUT INITIAL GAS GENERATOR SPEED,"NG".')
C
      READ(5,*) NG
      WRITE(6,*) NG
C

```


C

WRITE(6,3)

3 FORMAT(/,3X,'INPUT INITIAL DYNO SPEED,"NS".')

C

READ(5,*) NS

WRITE(6,*) NS

C

WRITE(9,*) ' NG=',NG

WRITE(9,*) ' NS=',NS

C

C

WRITE(6,4) NMF

4 FORMAT(/,3X,'INPUT NUMBER OF FUEL GUESSES,"NMF".')

READ(5,*) NMF

WRITE(6,*) NMF

C

C

WRITE(6,5) NP2

5 FORMAT(/,3X,'INPUT NUMBER OF P2 GUESSES,"NP2".')

READ(5,*) NP2

WRITE(6,*) NP2

C

C

WRITE(6,6) NP4

6 FORMAT(/,3X,'INPUT NUMBER OF P4 GUESSES,"NP4".')

READ(5,*) NP4

WRITE(6,*) NP4

C

WRITE(6,7) NRNO

7 FORMAT(/,3X,'INPUT THE NUMBER OF THIS REFINEMENT,"NRNO".')

READ(5,*) NRNO

WRITE(6,*) NRNO

C

C

VARIABLE INITIALIZATION

C

RNMF = NMF

RNP2 = NP2

RNP4 = NP4

QCONV = 10.0

P2CONV = 0.5

P4CONV = 0.5

MFLOW = 86.0000000

MFHIGH = 90.00000

P4LOW = 15.000000

P4HIGH = 16.0000000

P2LOW = 22.0000000

P2HIGH = 23.000000

ATEST = 1000.0

P2SAVC = 0.0

P2SAVG = 0.0

P4SAVC = 0.0

P4SAVG = 0.0

MFSAVG = 0.0

C

DMF = 40.0 / RNMF

```

DP2 = 25.0 / RNP2
DP4 = 5.0 / RNP4
MFL = MFLOW - DMF
WRITE(6,*) 'MFL=',MFL
NMF = (MFHIGH - MFL) / DMF
NNMF = NMF * 5
RNNMF = NNMF
DMF2 = DMF / 5.00
NNMF = NNMF + 10
C   WRITE(6,*) 'NNMF=',NNMF
C   WRITE(6,*) 'DMF2=',DMF2
C
P2L = P2LOW - DP2
C   WRITE(6,*) 'P2L=',P2L
NP2 = (P2HIGH - P2L) / DP2
NNP2 = NP2 * 10
RNNP2 = NNP2
DP22 = DP2 / 10.0
NNP2 = NNP2 + 10
C   WRITE(6,*) 'NNP2=',NNP2
C   WRITE(6,*) 'DP22=',DP22
P4H = P4HIGH + DP4
NP4 = (P4H - P4LOW) / DP4
NNP4 = NP4 * 10
RNNP4 = NNP4
DP42 = DP4 / 10.0
NNP4 = NNP4 + 10
C   WRITE(6,*) 'NNP4=',NNP4

```

```

C      WRITE(6,*) 'DP42=',DP42
C
C      REINITIALIZE  VARIABLES
C      DMF = (MFHIGH - MFLOW)/RNMF
C      DP2 = (P2HIGH - P2LOW) / RNP2
C      DP4 = (P4HIGH - P4LOW) / RNP4
C
C
C
C      MFHIGH = 20.0
C      MFLOW = 500.0
C      P2LOW = 300.0
C      P2HIGH = 1.0
C      P4LOW = 300.0
C      P4HIGH = 1.0
C
C      MFG = MFL - DMF2
C      MF LOOP
C      MFG = MFLOW - DMF
C
C      DO 100 J = 1,NNMF
C      DO 100 J = 1,NMF
C      MFG = MFG + DMF2
C      MFG = MFG + DMF
C      P2SAVE = 0.0
C      P2G = P2L - DP22
C      P2G = P2LOW - DP2
C      WRITE(6,*)

```

```

WRITE(6,*) 'MFG=',MFG
C
C      P2 LOOP
C
      DO 200 K = 1,NNP2
C      DO 200 K = 1,NP2
C      P2G = P2G + DP2
      P2G = P2G + DP22
      CALL SUBQC(NG,P2G,QC)
      CALL SUBT2(NG,P2G,T2)
      CALL SUBMA(NG,P2G,MA)
      QSAVE = 0.0
      P4SAVE = 0.0
      P4G = P4H + DP42
C      P4G = P4HIGH - DP4
C      WRITE(6,*) 'P4G=',P4G
C
C      P4 LOOP
C
      DO 300 L = 1,NNP4
      P4G = P4G - DP42
C      DO 300 L = 1,NP4
C      P4G = P4G + DP4
C
C      TORQUE CONVERGENCE
C
      CALL SUBQHT(NG,MA,T2,MFG,P4G,QHPT)
      DELQ = QC - QHPT

```

```

      QTEST = DELQ * QSAVE
      QSAVE = DELQ
C     WRITE(6,*) 'QTEST=',QTEST
      IF(QTEST.LT.0.0) GO TO 10
      8   GO TO 300
C     9   IF(ABS(DELQ).GT.QCONV) GO TO 300
C
C     P4 CONVERGENCE
C
      10  MAF = MA + MF
      CALL SUBT4(NG,MA,T2,MFG,P4G,T4)
      CALL SUBP4(MAF,T4,NS,P4C)
      DELP4 = P4C - P4G
      P4TEST = DELP4*P4SAVE
      P4SAVE = DELP4
C     WRITE(6,*) 'P4TEST=',P4TEST
      IF(P4TEST.LE.0.0) GO TO 20
      14  GO TO 300
C     15  IF(ABS(DELP4).GT.P4CONV) GO TO 300
C
C     P2 CONVERGENCE
C
      20  CALL SUBP2(NG,MA,T2,MFG,P4G,P2C)
      DELP2 = P2C - P2G
      P2TEST = DELP2*P2SAVE
      P2SAVE = DELP2
C     WRITE(6,*) 'P2TEST=',P2TEST
      IF(P2TEST.LE.0.0) GO TO 30

```

```

24      GO TO 300
C 25      IF(ABS(DELP2).GT.P2CONV) GO TO 300
30      IF(MFG.LT.MFLOW) MFLOW = MFG
        IF(MFG.GT.MFHIGH) MFHIGH = MFG
        IF(P4G.LT.P4LOW) P4LOW = P4G
        IF(P4G.GT.P4HIGH) P4HIGH = P4G
        IF(P2G.LT.P2LOW) P2LOW = P2G
        IF(P2G.GT.P2HIGH) P2HIGH = P2G
C      WRITE(6,*) 'CONVERGENCE HERE'
C
        DELSUM = (DELQ**2) + (DELP2**2) + (DELP4**2)
        IF(DELSUM.GE.ATEST) GO TO 300
35      P2SAVC = P2C
        P2SAVG = P2G
        P4SAVC = P4C
        P4SAVG = P4G
        MFSAVG = MFG
        DELQG = DELQ
        DELP2G = DELP2
        DELP4G = DELP4
        ATEST = DELSUM
C
C
        WRITE(6,*) ' CONVERGENCE OBTAINED AT'
        WRITE(6,*) ' P2C=',P2SAVC
        WRITE(6,*) ' P2G=',P2SAVG
        WRITE(6,*) ' P4C=',P4SAVC
        WRITE(6,*) ' P4G=',P4SAVG

```

```

WRITE(6,*) ' MF=' ,MFSAVG
WRITE(6,*) ' DELQ=' ,DELQG
WRITE(6,*) ' DELP2=' ,DELP2G
WRITE(6,*) ' DELP4=' ,DELP4G
WRITE(6,*)

```

C

```

WRITE(9,*)
WRITE(9,*) ' NUMBER OF REFINEMENTS=' ,NRNO
WRITE(9,*) ' CONVERGENCE OBTAINED AT'
WRITE(9,*) ' P2C=' ,P2SAVC
WRITE(9,*) ' P2G=' ,P2SAVG
WRITE(9,*) ' P4C=' ,P4SAVC
WRITE(9,*) ' P4G=' ,P4SAVG
WRITE(9,*) ' MF=' ,MFSAVG
WRITE(9,*) ' DELQ=' ,DELQG
WRITE(9,*) ' DELP2=' ,DELP2G
WRITE(9,*) ' DELP4=' ,DELP4G

```

C

```

WRITE(9,*)
WRITE(9,*) ' MFLOW=' ,MFLOW
WRITE(9,*) ' MFHIGH=' ,MFHIGH
WRITE(9,*) ' P4LOW=' ,P4LOW
WRITE(9,*) ' P4HIGH=' ,P4HIGH
WRITE(9,*) ' P2LOW=' ,P2LOW
WRITE(9,*) ' P2HIGH=' ,P2HIGH

```

C

```

P4 = P4G + DP42
CALL PART(A,B,MA,P2G,T2,MFG,NG,NS,P4G,T4,MAF,WW)

```

300 CONTINUE

C


```

200    CONTINUE
100    CONTINUE
C
      WRITE(6,*) '  CONVERGENCE OBTAINED AT'
      WRITE(6,*) '  P2C=',P2SAVC
      WRITE(6,*) '  P2G=',P2SAVG
      WRITE(6,*) '  P4C=',P4SAVC
      WRITE(6,*) '  P4G=',P4SAVG
      WRITE(6,*) '  MF=',MFSAVG
      WRITE(6,*) '  DELQ=',DELQG
      WRITE(6,*) '  DELP2=',DELP2G
      WRITE(6,*) '  DELP4=',DELP4G
C
C
      WRITE(6,*) '  MFLOW=',MFLOW
      WRITE(6,*) '  MFHIGH=',MFHIGH
      WRITE(6,*) '  P4LOW=',P4LOW
      WRITE(6,*) '  P4HIGH=',P4HIGH
      WRITE(6,*) '  P2LOW=',P2LOW
      WRITE(6,*) '  P2HIGH=',P2HIGH
C
C
C
C
900    STOP
      END

```

APPENDIX C

* * *

* ROUTINE SSVPD *

❁ ❁

* MODIFIED BY *

* V. A. STAMMETTI *

* FROM A SUBROUTINE BY *

* V. J. HERDA *

✻

* * *

* THIS ROUTINE IS THE THIRD AND FINAL OF A THREE STEP *

* PROCESS TO OBTAIN STEADY STATE CONVERGENCE AT A POINT *

* IN THE OPERATING ENVELOPE OF THE BOEING 501-6A GAS *

* TURBINE ENGINE INSTALLED AT THE NAVAL POSTGRADUATE *

* SCHOOL. ROUTINE SSVPD PROVIDES THE STEADY STATE *

* VALUES FOR ALL NECESSARY PLANT VARIABLES, AS WELL AS *

* THE STATE SPACE "A" AND "B" MATRICES. *

* * *

REAL*8 NG,NS,MF,MA,MAF,NSO,NGO,MFO,MAFO,MAO,MFDEL,MFU,MFL,

1 MIR,MF2,MFMIN,MERR,MAC,AFC,T2,T4,P4C,P2G,MFI,

```

1  LROOT,HROOT,P4G1,P4G2,P4TEST,AA,BB,CC,AA1,BB1,CC1,
1  SCALE,P2DEL,P4CONV,FO
    DIMENSION A(3,3),B(3,2)

C
C    INPUT THE INITIAL GAS GENERATOR SPEED AND DYNO SPEED.
C
    WRITE(6,2)
2  FORMAT(/,3X,'INPUT INITIAL GAS GENERATOR SPEED,"NG".')
C
    READ(5,*) NG
    WRITE(6,*) NG
C
C
    WRITE(6,3)
3  FORMAT(/,3X,'INPUT INITIAL DYNO SPEED,"NS".')
C
    READ(5,*) NS
    WRITE(6,*) NS
C
C
    WRITE(6,4) P2
4  FORMAT(/,3X,'INPUT PRESSURE,"P2".')
    READ(5,*) P2
    WRITE(6,*) P2
C
C
    WRITE(6,5) NP4
5  FORMAT(/,3X,'INPUT PRESSURE,"P4".')

```

```

      READ(5,*) P4
      WRITE(6,*) P4

C
      WRITE(6,6) MF
6     FORMAT(/,3X,'INPUT FUEL,"MF".')
      READ(5,*) MF
      WRITE(6,*) MF

C
C     PARAMETER CALCULATIONS
C
      CALL SUBMA(NG,P2,MA)
      CALL SUBT2(NG,P2,T2)
      CALL SUBT4(NG,MA,T2,MF,P4,T4)
      CALL SUBQC(NG,P2,QC)
      CALL SUBQHT(NG,MA,T2,MF,P4,QHPT)
      MAF = MA + MF
      CALL SUBQFT(MAF,T4,NS,QFPT)
      QD = QFPT
      C5 = 1.19294E-5
      C3 = 4.0E-6
      C4 = -20.0 + C3*NS*NS
      MIR = (QD - C4) / (C5*NS*NS)
      IF(MIR.LT.0.0) MIR = 0.0
      WW = MIR**(1.0/1.3)

C
C     COMPUTATION OF THE STATE SPACE MATRIX COEFFICIENTS
C
      CALL PART(A,B,MA,P2,T2,MF,NG,NS,P4,T4,MAF,WW)

```

C
C
C
C

```
WRITE(6,*) ' NG = ',NG
WRITE(6,*) ' NS = ',NS
WRITE(6,*) ' MF = ',MF
WRITE(6,*) ' P2 = ',P2
WRITE(6,*) ' P4 = ',P4
WRITE(6,*) ' T2 = ',T2
WRITE(6,*) ' T4 = ',T4
WRITE(6,*) ' MA = ',MA
WRITE(6,*) ' MAF = ',MAF
WRITE(6,*) ' QC = ',QC
WRITE(6,*) ' QHPT = ',QHPT
WRITE(6,*) ' QFPT = ',QFPT
WRITE(6,*) ' QD = ',QD
WRITE(6,*) ' WW = ',WW
```

C

```
WRITE(9,*) ' NG = ',NG
WRITE(9,*) ' NS = ',NS
WRITE(9,*) ' MF = ',MF
WRITE(9,*) ' P2 = ',P2
WRITE(9,*) ' P4 = ',P4
WRITE(9,*) ' T2 = ',T2
WRITE(9,*) ' T4 = ',T4
WRITE(9,*) ' MA = ',MA
WRITE(9,*) ' MAF = ',MAF
```

```

WRITE(9,*) ' QC = ',QC
WRITE(9,*) ' QHPT = ',QHPT
WRITE(9,*) ' QFPT = ',QFPT
WRITE(9,*) ' QD = ',QD
WRITE(9,*) ' WW = ',WW

C
C
900 STOP
END

C
C*****
SUBROUTINE PART(A,B,MA,P2,T2,MF,NG,NS,P4,T4,MAF,WW)
C*****
C
C THIS SUBROUTINE CALCULATES THE ELEMENTS OF THE 'A' AND 'B' MATRICES
C IN THE STATE SPACE EQUATION:
C
C 
$$\dot{X} = A \cdot X + B \cdot U$$

C
C
C COMMON QC,NG,P2,QH,MA,T2,MF,P4,QF,MAF,T4,NS,QD,WW
C DIMENSION A(3,3),B(3,2)
C REAL NG,NS,MF,MA,MAF,JG,JD,DENOM1,DENOM2,DENOM3
C
C
C
JG = 0.009525 * 2 * 3.14159 / 60.0
JD = 0.6738 * 2 * 3.14159 / 60.0

```

C

C CALL SUBROUTINES TO GET PARTIAL DERIVATIVES.

C

CALL DMA(NG,P2,DMADNG,DMADP2)

CALL DT2(NG,P2,DT2DNG,DT2DP2)

CALL DQC(NG,P2,DQCDNG,DQCDP2)

CALL DP2(NG,MA,T2,MF,P4,DP2DNG,DP2DMF,DP2DMA,DP2DT2,DP2DP4)

CALL DT4(NG,MA,T2,MF,P4,DT4DNG,DT4DMF,DT4DMA,DT4DT2,DT4DP4)

CALL DQHT(NG,MA,T2,MF,P4,DQHDNG,DQHDMF,DQHDMA,DQHDT2,DQHDP4)

CALL DP4(MAF,T4,NS,DP4DNS,DP4MAF,DP4DT4)

CALL DQFT(MAF,T4,NS,DQFDNS,DQFMAF,DQFDT4)

CALL DQD(NS,WW,DQDDNS,DQDDWW)

C

C COMPUTE THE COEFFICIENTS OF THE STATE SPACE EQUATIONS (I.E. THE
C ELEMENTS OF THE 'A' AND 'B' MATRICES).

C

J1 = DQHDMA

J2 = DQHDMF

J3 = DQHDT2

J4 = DQHDP4

J5 = DQHDNG

E1 = DQCDP2

E2 = DQCDNG

C1 = DMADP2

C2 = DMADNG

D1 = DT2DP2

D2 = DT2DNG

A1 = DP4MAF

$A2 = DP4DT4$
 $A3 = DP4DNS$
 $H1 = DP2DMA$
 $H2 = DP2DMF$
 $H3 = DP2DT2$
 $H4 = DP2DP4$
 $H5 = DP2DNG$
 $G1 = DT4DMA$
 $G2 = DT4DMF$
 $G3 = DT4DT2$
 $G4 = DT4DP4$
 $G5 = DT4DNG$
 $B1 = DQFMAF$
 $B2 = DQFDT4$
 $B3 = DQFDNS$
 $Z1 = B2 * G3 * D2 + B2 * G5$
 $Z2 = B1 + B2 * G2$
 $Z3 = B1 + B2 * G1$
 $Z4 = B2 * G3 * D1$
 $Z5 = B2 * G4$
 $Z6 = Z1 + Z3 * C2$
 $Z7 = Z4 + Z3 * C1$
 $DENOM1 = 1.0 - H1 * C1 - H3 * D1$
 $Y1 = (H5 + H1 * C2 + H3 * D2) / DENOM1$
 $Y2 = H2 / DENOM1$
 $Y3 = H4 / DENOM1$
 $DENOM2 = 1.0 - A2 * G4$
 $Y4 = (A2 * G5 + A1 * C2 + A2 * G3 * D2 + A2 * G1 * C2) / DENOM2$


```

Y5 = A3 / DENOM2
Y6 = (A1 + A2*G2) / DENOM2
Y7 = (A1*C1 + A2*G1*C1 + A2*G3*D1) / DENOM2
DENOM3 = 1.0 - Y7*Y3
Y8 = (Y4 + Y7*Y1) / DENOM3
Y9 = Y5 / DENOM3
Y10 = (Y6 + Y7*Y2) / DENOM3
Z8 = Z6 + Z7*Y1 + Z5*Y8 + Z7*Y3*Y8
Z9 = Z5*Y9 + Z7*Y3*Y9 + B3
Z10 = Z2 + Z7*Y2 + Z5*Y10 + Z7*Y3*Y10
Y11 = J5 - E2 + J1*C2 + J3*D2
Y12 = J1*C1 + J3*D1 - E1
Y13 = Y11 + Y12*Y1
Y14 = J2 + Y12*Y2
Y15 = J4 + Y12*Y3
Z11 = Y13 + Y15*Y8
Z12 = Y15*Y9
Z13 = Y14 + Y15*Y10

```

C

C FINAL FORM OF THE 'A' AND 'B' MATRICES.

C ! NOTE ! ELEMENTS A33 AND B31 ARE NOT COMPUTED HERE BUT WERE

C DETERMINED EXPERIMENTALLY FROM GAS TURBINE TEST DATA.

C

C FOR ACCELERATIONS USE:

C

C A33 = -0.5

C B31 = 0.5

C

C

C

C

C

B32 = 0.0

```
WRITE(9,*) '      A13 = ', A13
```

```
WRITE(9,*) '      A21 = ', A21
WRITE(9,*) '      A22 = ', A22
WRITE(9,*) '      A23 = ', A23
WRITE(9,*) '      A31 = ', A31
WRITE(9,*) '      A32 = ', A32
WRITE(9,*) '      A33 = ', A33
WRITE(9,*) '      B11 = ', B11
WRITE(9,*) '      B12 = ', B12
WRITE(9,*) '      B21 = ', B21
WRITE(9,*) '      B22 = ', B22
WRITE(9,*) '      B31 = ', B31
WRITE(9,*) '      B32 = ', B32
```

C

```
RETURN
```

```
END
```

APPENDIX D

```
*****
*
*   THE FOLLOWING SUBROUTINES WERE WRITTEN BY V. J. HERDA
*   AND ARE USED IN VARIOUS COMBINATIONS BY THE ROUTINES VRD,
*   SC, AND SSVPD.
*
*****
C
C
*****

      SUBROUTINE NGNSMF(X1,X2,BR)
C*****
C
C   THIS SUBROUTINE PRODUCES AN INITIAL "GOOD GUESS" FOR 'MF'
C   BASED ON THE SPECIFIED INPUTS, 'NG' AND 'NS'.
C
C
C
      DIMENSION X(5),C(21),Z(5),XR(5)
C
      XR(1) = X1
      XR(2) = X2
C
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
      C( 1)= 1.982237
      C( 2)= 0.2461511
```

C(3)= -5.147902E-02

C(4)= -1.884269

C(5)= -9.572456E-02

C(6)= 0.7639713

C

C SCALING FACTORS.

C

Z(1) = 36000.0

Z(2) = 3000.0

Z(3) = 240.0

C

NIND = 2

C

DO 686 I = 1,NIND

X(I) = XR(I)/Z(I)

686 CONTINUE

C

C CONSTRUCT THE COMPLETE QUADRATIC EQUATION.

C

B = 0

K = 0

DO 70 J = 1,NIND

DO 71 I = J,NIND

K = K+1

B = B+C(K)*X(J)*X(I)

71 CONTINUE

70 CONTINUE

C

```

DO 72 J = 1,NIND
    K = K+1
    B = B+C(K)*X(J)
72  CONTINUE
C
    K = K+1
    B = B+C(K)
C
C    WRITE(6,84) B
C 84  FORMAT(/,2X,'THE SCALED MF IS : ',2X,G15.7)
C
C
    BR = B * Z(NIND + 1)
C
C
C  THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
C    XHI = 240.0
C    XLO = 70.0
C
C    BR = AMAX1(XLO,BR)
C    BR = AMIN1(XHI,BR)
C
C    WRITE(6,85) BR
C 85  FORMAT(/,2X,'MF IS : ',2X,G15.7)
C
C
    RETURN

```

```

      END

C
C*****
      SUBROUTINE SUBMA(X1,X2,BR)
C*****
C
C THIS SUBROUTINE PRODUCES OUTPUT 'MA' FOR THE GIVEN INPUTS.
C
      DIMENSION X(5),C(21),Z(5),XR(5)
C
      XR(1) = X1
      XR(2) = X2
C
C
C COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
      C( 1)= 1.570198
      C( 2)= -0.7270151
      C( 3)= 0.2529498
      C( 4)= 0.1880112
      C( 5)= -0.6588774
      C( 6)= 0.3668176
C
C SCALING FACTORS.
C
      Z(1) = 36000.0
      Z(2) = 43.0
      Z(3) = 13000.0

```

```

C
      NIND = 2
C
      DO 686 I = 1,NIND
          X(I) = XR(I)/Z(I)
686      CONTINUE
C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
      B = 0
      K = 0
      DO 70 J = 1,NIND
          DO 71 I = J,NIND
              K = K+1
              B = B+C(K)*X(J)*X(I)
71      CONTINUE
70      CONTINUE
C
      DO 72 J = 1,NIND
          K = K+1
          B = B+C(K)*X(J)
72      CONTINUE
C
      K = K+1
      B = B+C(K)
C
C   WRITE(6,84) B
C 84  FORMAT(/,2X,'THE SCALED MA IS : ',2X,G15.7)

```



```

C
C
      BR = B * Z(NIND + 1)
C
C
C   THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
C      XHI = 13500.0
C      XLO = 5500.0
C      XHI = 15000.0
C      XLO = 4000.0
C
C      BR = AMAX1(XLO,BR)
C      BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C 85   FORMAT(/,2X,'MA IS : ',2X,G15.7)
C
C
      RETURN
      END
C
C*****
      SUBROUTINE SUBT2(X1,X2,BR)
C*****
C
C THIS SUBROUTINE PRODUCES OUTPUT 'T2' FOR THE GIVEN INPUTS.
C

```

C

DIMENSION X(5),C(21),Z(5),XR(5)

C

C

XR(1) = X1

XR(2) = X2

C

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C(1)= -0.5771397

C(2)= 2.203628

C(3)= -1.040498

C(4)= 0.1354878

C(5)= -0.4898891

C(6)= 0.7473461

C

C SCALING FACTORS.

C

Z(1) = 36000.0

Z(2) = 43.0

Z(3) = 800.0

C

NIND = 2

C

711 DO 500 I = 1,NIND

X(I) = XR(I)/Z(I)

500 CONTINUE

C

```

C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
      B = 0
      K = 0
      DO 70 J = 1,NIND
      DO 71 I = J,NIND
          K = K+1
          B = B+C(K)*X(J)*X(I)
71      CONTINUE
70      CONTINUE
C
      DO 72 J = 1,NIND
          K = K+1
          B = B+C(K)*X(J)
72      CONTINUE
C
          K = K+1
          B = B+C(K)
C
C      WRITE(6,84) B
C 84  FORMAT(/,2X,'THE SCALED T2 IS : ',2X,G15.7)
C
C
      BR = B * Z(NIND + 1)
C
C
C   THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.

```

```

C
C      XHI = 850.0
C      XLO = 500.0
C      XHI = 1000.0
C      XLO = 400.0
C
C      BR = AMAX1(XLO,BR)
C      BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C 85    FORMAT(/,2X,'T2 IS : ',2X,G15.7)
C
C
C      RETURN
C
C      END
C
C*****
C      SUBROUTINE SUBQC(X1,X2,BR)
C*****
C
C  THIS SUBROUTINE PRODUCES OUTPUT 'QC' FOR THE GIVEN INPUTS.
C
C      DIMENSION X(5),C(21),Z(5),XR(5)
C
C      XR(1) = X1
C      XR(2) = X2
C
C
C

```

C

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C(1)= -9.796132

C(2)= 20.03512

C(3)= -10.70980

C(4)= 0.1464243

C(5)= 1.657819

C(6)= -0.3884839

C

C SCALING FACTORS.

C

Z(1) = 36000.0

Z(2) = 43.0

Z(3) = 130.0

C

NIND = 2

C

711 DO 500 I = 1,NIND

X(I) = XR(I)/Z(I)

500 CONTINUE

C

C

C CONSTRUCT THE COMPLETE QUADRATIC EQUATION.

C

B = 0

K = 0

DO 70 J = 1,NIND

```

DO 71 I = J,NIND
      K = K+1
      B = B+C(K)*X(J)*X(I)
71  CONTINUE
70  CONTINUE
C
      DO 72 J = 1,NIND
            K = K+1
            B = B+C(K)*X(J)
72  CONTINUE
C
            K = K+1
            B = B+C(K)
C
C      WRITE(6,84) B
C 84  FORMAT(/,2X,'THE SCALED QC IS :',2X,G15.7)
C
C
      BR = B * Z(NIND + 1)
C
C
C  THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
C      XHI = 130.0
C      XLO = 40.0
C      XHI = 300.0
C      XLO = 0.0
C

```

```

C      BR = AMAX1(XLO,BR)
C      BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C 85   FORMAT(/,2X,'QC IS : ',2X,G15.7)
C
C
C      RETURN
C      END
C
C
C
C*****
C      SUBROUTINE SUBP2(X1,X2,X3,X4,X5,BR)
C*****
C
C  THIS SUBROUTINE PRODUCES OUTPUT 'P2' FOR THE GIVEN INPUTS.
C
C
C      DIMENSION X(5),C(21),Z(6),XR(5)
C
C
C      XR(1) = X1
C      XR(2) = X2
C      XR(3) = X3
C      XR(4) = X4
C      XR(5) = X5
C

```

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C(1)= 4.17287
C(2)= -2.839741
C(3)= 1.223014
C(4)= -1.854803
C(5)= -10.26167
C(6)= -0.2169524
C(7)= -1.156939
C(8)= 2.860795
C(9)= 5.767990
C(10)= -0.101891
C(11)= 0.2918
C(12)= -0.441640
C(13)= 0.7359644
C(14)= -9.559825
C(15)= 22.88968
C(16)= 4.7794
C(17)= -2.558953
C(18)= 0.272224
C(19)= 6.295503
C(20)= -28.57775
C(21)= 9.380198

C

C

C SCALING FACTORS.

C

Z(1) = 36000.0


```

      Z(2) =    13000.0
      Z(3) =     800.0
      Z(4) =     240.0
      Z(5) =     20.0
      Z(6) =     43.0

C

      NIND = 5

C

711    DO 500 I = 1,NIND
      X(I) = XR(I)/Z(I)

500    CONTINUE

C

C

C

C

C    CONSTRUCT THE COMPLETE QUADRATIC EQUATION.

C

      B = 0
      K = 0
      DO 70 J = 1,NIND
      DO 71 I = J,NIND
          K = K+1
          B = B+C(K)*X(J)*X(I)
71    CONTINUE
70    CONTINUE

C

      DO 72 J = 1,NIND
          K = K+1

```

```

        B = B+C(K)*X(J)
72      CONTINUE
C
        K = K+1
        B = B+C(K)
C
C      WRITE(6,84) B
C 84    FORMAT(/,2X,'THE SCALED P2 IS : ',2X,G15.7)
C
C
        BR = B * Z(NIND + 1)
C
C
C    THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
C      XHI = 43.0
C      XLO = 20.0
C      XHI = 100.0
C      XLO = 0.0
C
C      BR = AMAX1(XLO,BR)
C      BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C 85    FORMAT(/,2X,'P2 IS : ',2X,G15.7)
C
C
        RETURN

```

```

      END

C
C
C*****
      SUBROUTINE SUBT4(X1,X2,X3,X4,X5,BR)
C*****
C
C THIS SUBROUTINE PRODUCES OUTPUT 'T4' FOR THE GIVEN INPUTS.
C
C
      DIMENSION X(5),C(21),Z(6),XR(5)
C
C
      XR(1) = X1
      XR(2) = X2
      XR(3) = X3
      XR(4) = X4
      XR(5) = X5
C
C COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
C
      C( 1)= -22.20944
      C( 2)= 10.79398
      C( 3)= 21.99301
      C( 4)= 86.64350
      C( 5)= -208.0447
      C( 6)= 1.232848

```

C(7)= -12.46899
C(8)= -64.69914
C(9)= 180.0014
C(10)= -0.6479730
C(11)= -11.01693
C(12)= 20.21592
C(13)= 16.70037
C(14)= -121.1824
C(15)= 183.1548
C(16)= 138.3667
C(17)= -117.2714
C(18)= -18.03533
C(19)= 72.75989
C(20)= -229.4335
C(21)= 73.97864

C

C

C SCALING FACTORS.

C

Z(1) = 36000.0
Z(2) = 13000.0
Z(3) = 800.0
Z(4) = 240.0
Z(5) = 20.0
Z(6) = 1800.0

C

C

NIND = 5

```

C
711    DO 500 I = 1,NIND
      X(I) = XR(I)/Z(I)
500    CONTINUE
C
C
C
C
C    CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
      B = 0
      K = 0
      DO 70 J = 1,NIND
        DO 71 I = J,NIND
          K = K+1
          B = B+C(K)*X(J)*X(I)
71      CONTINUE
70      CONTINUE
C
      DO 72 J = 1,NIND
        K = K+1
        B = B+C(K)*X(J)
72      CONTINUE
C
      K = K+1
      B = B+C(K)
C
C    WRITE(6,84) B

```

```

C 84    FORMAT(/,2X,'THE SCALED T4 IS : ',2X,G15.7)
C
C
C      BR = B * Z(NIND + 1)
C
C
C      THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
C      XHI = 2000.0
C      XLO = 1300.0
C      XHI = 5000.0
C      XLO = 0.0
C
C      BR = AMAX1(XLO,BR)
C      BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C 85    FORMAT(/,2X,'T4 IS : ',2X,G15.7)
C
C
C      RETURN
C      END
C
C
C*****
C      SUBROUTINE SUBQHT(X1,X2,X3,X4,X5,BR)
C*****
C

```

C THIS SUBROUTINE PRODUCES OUTPUT 'QHPT' FOR THE GIVEN INPUTS.

C

C

DIMENSION X(5),C(21),Z(6),XR(5)

C

C

XR(1) = X1

XR(2) = X2

XR(3) = X3

XR(4) = X4

XR(5) = X5

C

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C(1)= 343.8178

C(2)= -562.3596

C(3)= -23.65895

C(4)= -54.97896

C(5)= 98.09515

C(6)= 217.8508

C(7)= 3.591497

C(8)= 119.9962

C(9)= -248.3938

C(10)= -0.1507291

C(11)= 17.95723

C(12)= -17.87346

C(13)= -40.67739

C(14)= 28.27711

C(15)= 190.2205
C(16)= -160.9423
C(17)= 260.8458
C(18)= 21.40023
C(19)= -34.85067
C(20)= -219.0661
C(21)= 62.16870

C

C

C SCALING FACTORS.

C

Z(1) = 36000.0
Z(2) = 13000.0
Z(3) = 800.0
Z(4) = 240.0
Z(5) = 20.0
Z(6) = 130.0

C

C

C

NIND = 5

C

711 DO 500 I = 1,NIND
X(I) = XR(I)/Z(I)
500 CONTINUE

C

C

C


```

C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
      B = 0
      K = 0
      DO 70 J = 1,NIND
      DO 71 I = J,NIND
          K = K+1
          B = B+C(K)*X(J)*X(I)
71      CONTINUE
70      CONTINUE
C
      DO 72 J = 1,NIND
          K = K+1
          B = B+C(K)*X(J)
72      CONTINUE
C
          K = K+1
          B = B+C(K)
C
C      WRITE(6,84) B
C 84  FORMAT(/,2X,'THE SCALED QHPT IS :',2X,G15.7)
C
C
      BR = B * Z(NIND + 1)
C
C
C   THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.

```

```

C
C      XHI = 130.0
C      XLO = 40.0
C      XHI = 300.0
C      XLO = 0.0
C
C      BR = AMAX1(XLO,BR)
C      BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C 85    FORMAT(/,2X,'QHPT IS : ',2X,G15.7)
C
C
C      RETURN
C
C      END
C
C*****
C      SUBROUTINE SUBP4(X1,X2,X3,BR)
C*****
C
C  THIS SUBROUTINE PRODUCES OUTPUT 'P4' FOR THE GIVEN INPUTS.
C
C
C      DIMENSION X(5),C(21),Z(6),XR(5)
C
C
C      XR(1) = X1
C      XR(2) = X2

```

XR(3) = X3

C

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C(1)= 0.1926178

C(2)= 1.158328

C(3)= 0.1008366

C(4)= 6.138049E-02

C(5)= 8.429369E-02

C(6)= -5.136141E-02

C(7)= -0.8789043

C(8)= -1.171511

C(9)= -4.834537E-02

C(10)= 1.559548

C

C

C SCALING FACTORS.

C

Z(1) = 13000.0

Z(2) = 1800.0

Z(3) = 3000.0

Z(4) = 20.0

C

C

C

C

NIND = 3

C

```

711      DO 500 I = 1,NIND
          X(I) = XR(I)/Z(I)
500      CONTINUE
C
C
C
C
C      CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
          B = 0
          K = 0
          DO 70 J = 1,NIND
              DO 71 I = J,NIND
                  K = K+1
                  B = B+C(K)*X(J)*X(I)
71          CONTINUE
70      CONTINUE
C
          DO 72 J = 1,NIND
              K = K+1
              B = B+C(K)*X(J)
72      CONTINUE
C
          K = K+1
          B = B+C(K)
C
C      WRITE(6,84) B
C 84      FORMAT(/,2X,'THE SCALED P4 IS :',2X,G15.7)

```

```

C
C
      BR = B * Z(NIND + 1)
C
C
C   THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
C      XHI = 20.0
C      XLO = 15.2
C      XHI = 50.0
C      XLO = 0.0
C
C      BR = AMAX1(XLO,BR)
C      BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C 85   FORMAT(/,2X,' P4 IS : ',2X,G15.7)
C
C
      RETURN
      END
C
C*****
      SUBROUTINE SUBQFT(X1,X2,X3,BR)
C*****
C
C THIS SUBROUTINE PRODUCES OUTPUT 'QFT' FOR THE GIVEN INPUTS.
C

```

C

DIMENSION X(5),C(21),Z(6),XR(5)

C

C

XR(1) = X1

XR(2) = X2

XR(3) = X3

C

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C

C

C(1)= 2.192477

C(2)= 0.8755642

C(3)= -0.6626919

C(4)= 3.892829

C(5)= -0.1769417

C(6)= 1.446682E-02

C(7)= -1.83825

C(8)= -7.607660

C(9)= 0.2095135

C(10)= 3.747696

C

C SCALING FACTORS.

C

Z(1) = 13000.0

Z(2) = 1800.0

Z(3) = 3000.0

```

      Z(4) = 480.00
C
C
C
      NIND = 3
C
711   DO 500 I = 1,NIND
      X(I) = XR(I)/Z(I)
500   CONTINUE
C
C
C   CONSTRUCT THE COMPLETE QUADRATIC EQUATION.
C
      B = 0
      K = 0
      DO 70 J = 1,NIND
      DO 71 I = J,NIND
          K = K+1
          B = B+C(K)*X(J)*X(I)
71   CONTINUE
70   CONTINUE
C
      DO 72 J = 1,NIND
          K = K+1
          B = B+C(K)*X(J)
72   CONTINUE
C
      K = K+1

```

```

      B = B+C(K)
C
C      WRITE(6,84) B
C 84  FORMAT(/,2X,'THE SCALED QFPT IS : ',2X,G15.7)
C
C
      BR = B * Z(NIND + 1)
C
C
C      THE FOLLOWING ENSURES THAT THE OUTPUT STAYS IN WITHIN LIMITS.
C
C      XHI = 480.0
C      XLO = 25.0
C
C      BR = AMAX1(XLO,BR)
C      BR = AMIN1(XHI,BR)
C
C      WRITE(6,85) BR
C 85  FORMAT(/,2X,'QFPT IS : ',2X,G15.7)
C
C
      RETURN
      END
C
C*****
      SUBROUTINE PART(A,B,MA,P2,T2,MF,NG,NS,P4,T4,MAF,WW)
C*****
C

```


C THIS SUBROUTINE CALCULATES THE ELEMENTS OF THE 'A' AND 'B' MATRICES
C IN THE STATE SPACE EQUATION:

C

C $\dot{X} = A \cdot X + B \cdot U$.

C

C

C COMMON QC,NG,P2,QH,MA,T2,MF,P4,QF,MAF,T4,NS,QD,WW

DIMENSION A(3,3),B(3,2)

REAL NG,NS,MF,MA,MAF,JG,JD,DENOM1,DENOM2,DENOM3

C

C

C

JG = 0.009525 * 2 * 3.14159 / 60.0

JD = 0.6738 * 2 * 3.14159 / 60.0

C

C CALL SUBROUTINES TO GET PARTIAL DERIVATIVES.

C

CALL DMA(NG,P2,DMADNG,DMADP2)

CALL DT2(NG,P2,DT2DNG,DT2DP2)

CALL DQC(NG,P2,DQCDNG,DQCDP2)

CALL DP2(NG,MA,T2,MF,P4,DP2DNG,DP2DMF,DP2DMA,DP2DT2,DP2DP4)

CALL DT4(NG,MA,T2,MF,P4,DT4DNG,DT4DMF,DT4DMA,DT4DT2,DT4DP4)

CALL DQHT(NG,MA,T2,MF,P4,DQHDNG,DQHDMF,DQHDMA,DQHDT2,DQHDP4)

CALL DP4(MAF,T4,NS,DP4DNS,DP4MAF,DP4DT4)

CALL DQFT(MAF,T4,NS,DQFDNS,DQFMAF,DQFDT4)

CALL DQD(NS,WW,DQDDNS,DQDDWW)

C

C COMPUTE THE COEFFICIENTS OF THE STATE SPACE EQUATIONS (I.E. THE

C ELEMENTS OF THE 'A' AND 'B' MATRICES).

C

J1 = DQHDMA

J2 = DQHDMF

J3 = DQHDT2

J4 = DQHDP4

J5 = DQHDNG

E1 = DQCDP2

E2 = DQCDNG

C1 = DMADP2

C2 = DMADNG

D1 = DT2DP2

D2 = DT2DNG

A1 = DP4MAF

A2 = DP4DT4

A3 = DP4DNS

H1 = DP2DMA

H2 = DP2DMF

H3 = DP2DT2

H4 = DP2DP4

H5 = DP2DNG

G1 = DT4DMA

G2 = DT4DMF

G3 = DT4DT2

G4 = DT4DP4

G5 = DT4DNG

B1 = DQFMAF

B2 = DQFDT4

$B3 = DQFDNS$
 $Z1 = B2 * G3 * D2 + B2 * G5$
 $Z2 = B1 + B2 * G2$
 $Z3 = B1 + B2 * G1$
 $Z4 = B2 * G3 * D1$
 $Z5 = B2 * G4$
 $Z6 = Z1 + Z3 * C2$
 $Z7 = Z4 + Z3 * C1$
 $DENOM1 = 1.0 - H1 * C1 - H3 * D1$
 $Y1 = (H5 + H1 * C2 + H3 * D2) / DENOM1$
 $Y2 = H2 / DENOM1$
 $Y3 = H4 / DENOM1$
 $DENOM2 = 1.0 - A2 * G4$
 $Y4 = (A2 * G5 + A1 * C2 + A2 * G3 * D2 + A2 * G1 * C2) / DENOM2$
 $Y5 = A3 / DENOM2$
 $Y6 = (A1 + A2 * G2) / DENOM2$
 $Y7 = (A1 * C1 + A2 * G1 * C1 + A2 * G3 * D1) / DENOM2$
 $DENOM3 = 1.0 - Y7 * Y3$
 $Y8 = (Y4 + Y7 * Y1) / DENOM3$
 $Y9 = Y5 / DENOM3$
 $Y10 = (Y6 + Y7 * Y2) / DENOM3$
 $Z8 = Z6 + Z7 * Y1 + Z5 * Y8 + Z7 * Y3 * Y8$
 $Z9 = Z5 * Y9 + Z7 * Y3 * Y9 + B3$
 $Z10 = Z2 + Z7 * Y2 + Z5 * Y10 + Z7 * Y3 * Y10$
 $Y11 = J5 - E2 + J1 * C2 + J3 * D2$
 $Y12 = J1 * C1 + J3 * D1 - E1$
 $Y13 = Y11 + Y12 * Y1$
 $Y14 = J2 + Y12 * Y2$

$Y15 = J4 + Y12*Y3$
 $Z11 = Y13 + Y15*Y8$
 $Z12 = Y15*Y9$
 $Z13 = Y14 + Y15*Y10$

C

C FINAL FORM OF THE 'A' AND 'B' MATRICES.

C ! NOTE ! ELEMENTS A33 AND B31 ARE NOT COMPUTED HERE BUT WERE
C DETERMINED EXPERIMENTALLY FROM GAS TURBINE TEST DATA.

C

C FOR ACCELERATIONS USE:

C

C $A33 = -0.5$

C $B31 = 0.5$

C

C FOR DECELERATIONS USE:

C

C $A33 = -0.87$

C $B31 = 0.87$

C

$A11 = Z11/JG$

$A12 = Z12/JG$

$A13 = Z13/JG$

$A21 = Z8/JD$

$A22 = Z9/JD$

$A23 = Z10/JD$

$A31 = 0.0$

$A32 = 0.0$

$A33 = -10.0$

B11 = 0.0

B12 = 0.0

B21 = 0.0

B22 = -1.0 / JD

B31 = 10.0

B32 = 0.0

C

C IF(A12.LT.0.0) RETURN

C IF(A13.LT.0.0) RETURN

C IF(A21.LT.0.0) RETURN

C IF(A23.LT.0.0) RETURN

C IF(A11.GT.0.0) RETURN

C IF(A22.GT.0.0) RETURN

WRITE(9,*) '"A" AND "B" MATRICES FOR NG = ',NG

WRITE(9,*) ' AND NS = ',NS

WRITE(9,*) ' '

WRITE(9,*) ' '

WRITE(9,*) ' A11 = ', A11

WRITE(9,*) ' A12 = ', A12

WRITE(9,*) ' A13 = ', A13

WRITE(9,*) ' A21 = ', A21

WRITE(9,*) ' A22 = ', A22

WRITE(9,*) ' A23 = ', A23

WRITE(9,*) ' A31 = ', A31

WRITE(9,*) ' A32 = ', A32

WRITE(9,*) ' A33 = ', A33

WRITE(9,*) ' B11 = ', B11

WRITE(9,*) ' B12 = ', B12

```

WRITE(9,*) '      B21 = ', B21
WRITE(9,*) '      B22 = ', B22
WRITE(9,*) '      B31 = ', B31
WRITE(9,*) '      B32 = ', B32

```

C

```

WRITE(6,*) '      A11 = ', A11
WRITE(6,*) '      A12 = ', A12
WRITE(6,*) '      A13 = ', A13
WRITE(6,*) '      A21 = ', A21
WRITE(6,*) '      A22 = ', A22
WRITE(6,*) '      A23 = ', A23

```

C

```

RETURN
END

```

C

C*****

```

SUBROUTINE DMA(X1,X2,DMADNG,DMADP2)

```

C*****

C

C

C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:

C

C

```

DMA/DNG, DMA/DP2

```

C

C

```

DIMENSION X(5),C(21),Z(5),XR(5)

```

C

```

XR(1) = X1

```

```

      XR(2) = X2
C
C
C   COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
      C( 1)=   1.570198
      C( 2)=  -0.7270151
      C( 3)=   0.2529498
      C( 4)=   0.1880112
      C( 5)=  -0.6588774
      C( 6)=   0.3668176
C
C   SCALING FACTORS.
C
      Z(1) =   36000.0
      Z(2) =    43.0
      Z(3) =   13000.0
C
      NIND = 2
C
      DO 686 I = 1,NIND
          X(I) = XR(I)/Z(I)
686   CONTINUE
C
C
      DMADNG = 2.0*C(1)*X(1) + C(2)*X(2) + C(4)
      DMADNG = DMADNG*Z(3)/Z(1)
C

```

```

DMADP2 = C(2)*X(1) + 2.0*C(3)*X(2) + C(5)
DMADP2 = DMADP2*Z(3)/Z(2)

C
C
C

RETURN

END

C
C*****
SUBROUTINE DT2(X1,X2,DT2DNG,DT2DP2)
C*****
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C          DT2/DNG, DT2/DP2
C
C
C
C          DIMENSION X(5),C(21),Z(5),XR(5)
C
C
C          XR(1) = X1
C          XR(2) = X2
C
C COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
C          C( 1)= -0.5771397

```


C(2)= 2.203628
C(3)= -1.040498
C(4)= 0.1354878
C(5)= -0.4898891
C(6)= 0.7473461

C

C SCALING FACTORS.

C

Z(1) = 36000.0
Z(2) = 43.0
Z(3) = 800.0

C

NIND = 2

C

711 DO 500 I = 1,NIND
X(I) = XR(I)/Z(I)
500 CONTINUE

C

C

C

DT2DNG = 2.0*C(1)*X(1) + C(2)*X(2) + C(4)
DT2DNG = DT2DNG*Z(3)/Z(1)

C

DT2DP2 = C(2)*X(1) + 2.0*C(3)*X(2) + C(5)
DT2DP2 = DT2DP2*Z(3)/Z(2)

C

C

RETURN

END

C

C*****

SUBROUTINE DQC(X1,X2,DQCDNG,DQCDP2)

C*****

C

C

C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:

C

C

DQC/DNG, DQC/DP2

C

C

DIMENSION X(5),C(21),Z(5),XR(5)

C

XR(1) = X1

XR(2) = X2

C

C

C

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C(1)= -9.796132

C(2)= 20.03512

C(3)= -10.70980

C(4)= 0.1464243

C(5)= 1.657819

C(6)= -0.3884839

C

C SCALING FACTORS.

C

Z(1) = 36000.0

Z(2) = 43.0

Z(3) = 130.0

C

NIND = 2

C

711 DO 500 I = 1,NIND

X(I) = XR(I)/Z(I)

500 CONTINUE

C

C

DQCDNG = 2.0*C(1)*X(1) + C(2)*X(2) + C(4)

DQCDNG = DQCDNG*Z(3)/Z(1)

C

DQCDP2 = C(2)*X(1) + 2.0*C(3)*X(2) + C(5)

DQCDP2 = DQCDP2*Z(3)/Z(2)

C

C

RETURN

END

C

C

C

C*****

SUBROUTINE DP2(X1,X2,X3,X4,X5,DP2DNG,DP2DMF,DP2DMA,DP2DT2,DP2DP4)

C*****

C

C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:

C

C

DP2/DNG, DP2/DMF

C

C

DIMENSION X(5),C(21),Z(6),XR(5)

C

C

XR(1) = X1

XR(2) = X2

XR(3) = X3

XR(4) = X4

XR(5) = X5

C

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C(1)= 4.17287

C(2)= -2.839741

C(3)= 1.223014

C(4)= -1.854803

C(5)= -10.26167

C(6)= -0.2169524

C(7)= -1.156939

C(8)= 2.860795

C(9)= 5.767990

C(10)= -0.101891

C(11)= 0.2918

C(12)= -0.441640

C(13)= 0.7359644

C(14)= -9.559825

C(15)= 22.88968

C(16)= 4.7794

C(17)= -2.558953

C(18)= 0.272224

C(19)= 6.295503

C(20)= -28.57775

C(21)= 9.380198

C

C

C SCALING FACTORS.

C

Z(1) = 36000.0

Z(2) = 13000.0

Z(3) = 800.0

Z(4) = 240.0

Z(5) = 20.0

Z(6) = 43.0

C

NIND = 5

C

711 DO 500 I = 1,NIND

X(I) = XR(I)/Z(I)

500 CONTINUE

C

C

DP2DNG = 2*C(1)*X(1) + C(2)*X(2) + C(3)*X(3) + C(4)*X(4)
 1 + C(5)*X(5) + C(16)

DP2DNG = DP2DNG*Z(6)/Z(1)

C

DP2DMF = C(4)*X(1) + C(8)*X(2) + C(11)*X(3) + 2*C(13)*X(4)
 1 + C(14)*X(5) + C(19)

DP2DMF = DP2DMF*Z(6)/Z(4)

C

DP2DMA = C(2)*X(1) + C(7)*X(3) + C(8)*X(4) + 2*C(6)*X(2)
 1 + C(9)*X(5) + C(17)

DP2DMA = DP2DMA*Z(6)/Z(2)

C

DP2DT2 = C(3)*X(1) + C(7)*X(2) + C(11)*X(4) + 2*C(10)*X(3)
 1 + C(12)*X(5) + C(18)

DP2DT2 = DP2DT2*Z(6)/Z(3)

C

DP2DP4 = C(5)*X(1) + C(9)*X(2) + C(12)*X(3) + 2*C(15)*X(5)
 1 + C(14)*X(4) + C(20)

DP2DP4 = DP2DP4*Z(6)/Z(5)

C

C

C

C

RETURN

END

C

C

C*****

```

SUBROUTINE DT4(X1,X2,X3,X4,X5,DT4DNG,DT4DMF,DT4DMA,DT4DT2,DT4DP4)
C*****
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C
C          DT4/DNG, DT4/DMF
C
C
C
C
C          DIMENSION X(5),C(21),Z(6),XR(5)
C
C
C          XR(1) = X1
C          XR(2) = X2
C          XR(3) = X3
C          XR(4) = X4
C          XR(5) = X5
C
C
C COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
C
C          C( 1)= -22.20944
C          C( 2)= 10.79398
C          C( 3)= 21.99301
C          C( 4)= 86.64350
C          C( 5)= -208.0447
C          C( 6)= 1.232848

```

C(7)= -12.46899
C(8)= -64.69914
C(9)= 180.0014
C(10)= -0.6479730
C(11)= -11.01693
C(12)= 20.21592
C(13)= 16.70037
C(14)= -121.1824
C(15)= 183.1548
C(16)= 138.3667
C(17)= -117.2714
C(18)= -18.03533
C(19)= 72.75989
C(20)= -229.4335
C(21)= 73.97864

C

C

C SCALING FACTORS.

C

Z(1) = 36000.0
Z(2) = 13000.0
Z(3) = 800.0
Z(4) = 240.0
Z(5) = 20.0
Z(6) = 1800.0

C

C

NIND = 5

C

711 DO 500 I = 1,NIND

X(I) = XR(I)/Z(I)

500 CONTINUE

C

C

DT4DNG = 2*C(1)*X(1) + C(2)*X(2) + C(3)*X(3) + C(4)*X(4)

1 + C(5)*X(5) + C(16)

DT4DNG = DT4DNG*Z(6)/Z(1)

C

DT4DMF = C(4)*X(1) + C(8)*X(2) + C(11)*X(3) + 2*C(13)*X(4)

1 + C(14)*X(5) + C(19)

DT4DMF = DT4DMF*Z(6)/Z(4)

C

DT4DMA = C(2)*X(1) + C(7)*X(3) + C(8)*X(4) + 2*C(6)*X(2)

1 + C(9)*X(5) + C(17)

DT4DMA = DT4DMA*Z(6)/Z(2)

C

DT4DT2 = C(3)*X(1) + C(7)*X(2) + C(11)*X(4) + 2*C(10)*X(3)

1 + C(12)*X(5) + C(18)

DT4DT2 = DT4DT2*Z(6)/Z(3)

C

DT4DP4 = C(5)*X(1) + C(9)*X(2) + C(12)*X(3) + 2*C(15)*X(5)

1 + C(14)*X(4) + C(20)

DT4DP4 = DT4DP4*Z(6)/Z(5)

C

C

C

```

C
      RETURN
      END
C
C
*****SSM15510
      SUBROUTINE
X2,X3,X4,X5,DQHDNG,DQHDMF,DQHDMA,DQHDT2,DQHDP4)SSM15520
*****SSM15530
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C      DQH/DNG, DQH/DMF, DQH/DMA, DQH/DT2, DQH/DP4
C
C
C
      DIMENSION X(5),C(21),Z(6),XR(5)
C
C
      XR(1) = X1
      XR(2) = X2
      XR(3) = X3
      XR(4) = X4
      XR(5) = X5
C
C COEFFICIENTS OF THE QUADRATIC CURVE FIT.
C
      C( 1)= 343.8178

```

C(2)= -562.3596
C(3)= -23.65895
C(4)= -54.97896
C(5)= 98.09515
C(6)= 217.8508
C(7)= 3.591497
C(8)= 119.9962
C(9)= -248.3938
C(10)= -0.1507291
C(11)= 17.95723
C(12)= -17.87346
C(13)= -40.67739
C(14)= 28.27711
C(15)= 190.2205
C(16)= -160.9423
C(17)= 260.8458
C(18)= 21.40023
C(19)= -34.85067
C(20)= -219.0661
C(21)= 62.16870

C

C

C SCALING FACTORS.

C

Z(1) = 36000.0
Z(2) = 13000.0
Z(3) = 800.0
Z(4) = 240.0

```

      Z(5) =    20.0
      Z(6) =   130.0

C
C
C
      NIND = 5

C
711   DO 500 I = 1,NIND
      X(I) = XR(I)/Z(I)
500   CONTINUE

C
C
C
      DQHDNG = 2*C(1)*X(1) + C(2)*X(2) + C(3)*X(3) + C(4)*X(4)
1      + C(5)*X(5) + C(16)
      DQHDNG = DQHDNG*Z(6)/Z(1)

C
      DQHDMF = C(4)*X(1) + C(8)*X(2) + C(11)*X(3) + 2*C(13)*X(4)
1      + C(14)*X(5) + C(19)
      DQHDMF = DQHDMF*Z(6)/Z(4)

C
      DQHDMA = C(2)*X(1) + C(7)*X(3) + C(8)*X(4) + 2*C(6)*X(2)
1      + C(9)*X(5) + C(17)
      DQHDMA = DQHDMA*Z(6)/Z(2)

C
      DQHDT2 = C(3)*X(1) + C(7)*X(2) + C(11)*X(4) + 2*C(10)*X(3)
1      + C(12)*X(5) + C(18)
      DQHDT2 = DQHDT2*Z(6)/Z(3)

```

```

C
      DQHDP4 = C(5)*X(1) + C(9)*X(2) + C(12)*X(3) + 2*C(15)*X(5)
1      + C(14)*X(4) + C(20)
      DQHDP4 = DQHDP4*Z(6)/Z(5)
C
C
      RETURN
      END
C
C*****
      SUBROUTINE DP4(X1,X2,X3,DP4DNS,DP4MAF,DP4DT4)
C*****
C
C
C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:
C
C
C      DP4/DNS
C
C
C
C      DIMENSION X(5),C(21),Z(6),XR(5)
C
C
C      XR(1) = X1
C      XR(2) = X2
C      XR(3) = X3
C
C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

```

C

C(1)= 0.1926178
C(2)= 1.158328
C(3)= 0.1008366
C(4)= 6.138049E-02
C(5)= 8.429369E-02
C(6)= -5.136141E-02
C(7)= -0.8789043
C(8)= -1.171511
C(9)= -4.834537E-02
C(10)= 1.559548

C

C

C SCALING FACTORS.

C

Z(1) = 13000.0
Z(2) = 1800.0
Z(3) = 3000.0
Z(4) = 20.0

C

C

C

C

NIND = 3

C

711 DO 500 I = 1,NIND
X(I) = XR(I)/Z(I)
500 CONTINUE

C

C

DP4DNS = C(3)*X(1) + C(5)*X(2) + 2*C(6)*X(3) + C(9)

DP4DNS = DP4DNS*Z(4)/Z(3)

C

DP4MAF = C(2)*X(2) + C(3)*X(3) + 2*C(1)*X(1) + C(7)

DP4MAF = DP4MAF*Z(4)/Z(1)

C

DP4DT4 = C(2)*X(1) + C(5)*X(3) + 2*C(4)*X(2) + C(8)

DP4DT4 = DP4DT4*Z(4)/Z(2)

C

C

RETURN

END

C

C*****

SUBROUTINE DQFT(X1,X2,X3,DQFDNS,DQFMAF,DQFDT4)

C*****

C

C

C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:

C

C

DQF/DNS, DQF/DMAF, DQF/DT4

C

C

C

DIMENSION X(5),C(21),Z(6),XR(5)

C

C

XR(1) = X1

XR(2) = X2

XR(3) = X3

C

C COEFFICIENTS OF THE QUADRATIC CURVE FIT.

C

C

C

C(1)= 2.192477

C(2)= 0.8755642

C(3)= -0.6626919

C(4)= 3.892829

C(5)= -0.1769417

C(6)= 1.446682E-02

C(7)= -1.83825

C(8)= -7.607660

C(9)= 0.2095135

C(10)= 3.747696

C

C SCALING FACTORS.

C

Z(1) = 13000.0

Z(2) = 1800.0

Z(3) = 3000.0

Z(4) = 480.00

C

C


```

C
      NIND = 3
C
711   DO 500 I = 1,NIND
      X(I) = XR(I)/Z(I)
500   CONTINUE
C
C
C
      DQFDNS = C(3)*X(1) + C(5)*X(2) + 2*C(6)*X(3) + C(9)
      DQFDNS = DQFDNS*Z(4)/Z(3)
C
      DQFMAF = C(2)*X(2) + C(3)*X(3) + 2*C(1)*X(1) + C(7)
      DQFMAF = DQFMAF*Z(4)/Z(1)
C
      DQFDT4 = C(2)*X(1) + C(5)*X(3) + 2*C(4)*X(2) + C(8)
      DQFDT4 = DQFDT4*Z(4)/Z(2)
C
C
C
      RETURN
      END
C
C*****
      SUBROUTINE DQD(X1,X2,DQDDNS,DQDDWW)
C*****
C
C

```

C THIS SUBROUTINE PRODUCES THE FOLLOWING PARTIAL DERIVATIVES:

C

C

DQD/DNS, DQD/DWW

C

C

C

SCALING FACTORS

C

XQD = 480.

XNS = 3000.

XWW = 49.

C

C1 = -20.0

C2 = 4.0E-6

C3 = 1.19294E-5

C

DQDDNS = 2*X1*C2 + 2*C3*(X2**1.3)*X1

C

DQDDWW = 1.3*C3*X1*X1*(X2**0.3)

C

C

DQDDNS = DQDDNS*XNS/XQD

C

DQDDWW = DQDDWW*XWW/XQD

C

C

C

RETURN

END

APPENDIX E

TITLE GT DYNAMIC PROGRAM

*

*

*

*

*

*

*

BOEING MODEL 502-6A GAS TURBINE

*

*

*

*

DYNAMIC COMPUTER SIMULATION

*

*

*

*

MODIFIED BY

*

*

V. A. STAMMETTI

*

*

FROM A ROUTINE BY

*

*

V. J. HERDA

*

*

*

*

THIS PROGRAM SIMULATES THE DYNAMIC RESPONSE OF THE NPS

*

*

BOEING GAS TURBINE TEST FACILITY USING A MULTIPLE

*

*

LINEARIZATION TECHNIQUE.

*

*

*

C

C

PARAM JG=0.009525, JD=0.6738, PI=3.14159, T = 2.0 , TW=2.0 , TW1=1.0

*
 * THE FOLLOWING VALUES LISTED ON THE FUNCTION CARD ARE FOR FUEL FLOW,
 * GAS GENERATOR SPEED, AND DYNO SPEED AS A FUNCTION OF TIME.
 * THESE VALUES WERE OBTAINED FROM STRIP CHART RECORDS AND ARE ENTERED
 * IN THE FORM (E.G. FUEL FLOW) ...TIME(SEC), FUEL FLOW.....

*
 * THIS SET IS FOR EXPERIMENTAL RUN # 1.

C

C THIS SET IS FOR EXPERIMENTAL RUN # 4.

C

C

AFGEN NGDATA= 0.0,34900.0, 5.0,34900.0, 10.0,34900.0, 15.0,34900.0, ...
 20.0,34900.0, 25.0,34900.0, 30.0,34900.0, 35.0,34900.0

AFGEN NSDATA= 0.0,570.0, 1.0,570.0, 3.0,570.0, 5.0,597.67, ...
 6.0,653.0, 7.0,708.4, 8.0,791.4, 9.0,846.7, 10.0,929.8, ...
 11.0,1040.5, 12.0,1178.8, 13.0,1344.9, 14.0,1566.3, ...
 15.0,1787.7, 16.0,2009.1, 17.0,2230.5, 18.0,2451.9, ...
 19.0,2590.2, 20.0,2673.3, 21.0,2756.3, 22.0,2839.3, ...
 23.0,2894.65, 24.0,2950.0, 25.0,2950.0, 26.0,2950.0, ...
 35.0,2950.0

AFGEN MFDATA= 0.0,193.5, 4.0,193.5, 5.0,195.6, 6.0,197.7, ...
 7.0,197.7, 8.0,197.7, 9.0,199.75, 10.0,199.75, ...
 11.0,199.75, 12.0,201.8, 13.0,201.8, 14.0,203.9, ...
 15.0,203.9, 16.0,206.0, 20.0,206.0, 25.0,206.0, ...
 30.0,206.0, 35.0,206.0

AFGEN QDDATA= 0.0,450.0, 4.0,450.0, 5.0,445.1, 6.0,445.1, ...
 7.0,435.4, 8.0,425.6, 9.0,415.8, 10.0,406.1, ...
 11.0,391.1, 12.0,376.8, 13.0,362.1, 14.0,332.9, ...

15.0,313.3, 16.0,288.9, 17.0,274.3, 18.0,264.5, ...
19.0,259.6, 20.0,254.8, 21.0,245.0, 25.0,245.0, ...
30.0,245.0, 35.0,245.0

*

*

INITIAL

*

* ESTABLISH INITIAL CONDITIONS.

*

NGI=34900.0

NSI=570.0

MFI = 193.5

QDI = 450.0

*

* ESTABLISH FINAL CONDITIONS

*

NGF=34900.0

NSF=2950.0

QDF = 206.00

MFF = 245.0

*

* SET INITIAL STATE PERTURBATION TO ZERO

*

DNG = 0.0

DNS = 0.0

DE = 0.0

*

EO = MFI

*

NGB = 26000.0

NSB = 1750.0

*

*

DYNAMIC

*

*

DATA CURVE FORMULATION

*

NGD = AFGEN(NGDATA,TIME)

NSD = AFGEN(NSDATA,TIME)

MFD = AFGEN(MFDATA,TIME)

QDD = AFGEN(QDDATA,TIME)

*

*

STATE SPACE LINEAR MODEL FORMULATION

*

A11 = -1.0*EXP(-6.9929E-01*(NSL/NSB) + 5.5831E+00*(NGL/NGB) ...
-3.2433E+00)

*

A12 = -1.0*EXP(-2.8415E+00*(NSL/NSB) + 7.9978E+00*(NGL/NGB) ...
-4.4662E+00)

*

*

A13 = (2.1503E+03)*((NSL/NSB)**2.0) ...

*

+ (-5.9752E+03)*(NSL/NSB)*(NGL/NGB) ...

*

+ (-5.0697E+03)*((NGL/NGB)**2.0) + (1.3101E+03)*(NSL/NSB) ...

*

+ (1.8551E+04)*(NGL/NGB) - 9.1460E+03

A13 = EXP(-0.45788*(NSL/NSB) + 1.189*(NGL/NGB) +6.8305)

*

$A_{21} = (1.5312E-01)*((NSL/NSB)**2.0) \dots$
 $+ (-9.5351E-01)*(NSL/NSB)*(NGL/NGB) \dots$
 $+ (1.5745E+00)*((NGL/NGB)**2.0) + (5.1810E-01)*(NSL/NSB) \dots$
 $+ (-1.6232E+00)*(NGL/NGB) + 4.6015E-01$

*

$A_{22} = (5.6875E-02)*(NSL/NSB) + (-1.3166E+00)*(NGL/NGB) \dots$
 $+ 3.9862E-01$

*

$A_{23} = (-1.7434E+01)*((NSL/NSB)**2.0) \dots$
 $+ (3.5345E+01)*(NSL/NSB)*(NGL/NGB) \dots$
 $+ (4.5787+01)*((NGL/NGB)**2.0) + (1.0894E+01)*(NSL/NSB) \dots$
 $+ (-2.0510E+02)*(NGL/NGB) + 1.5265E+02$
 $A_{23} = \text{EXP}(0.92011*(NSL/NSB) - 4.2549*(NGL/NGB) + 6.2290)$

*

$A_{31} = 0.0$

*

$A_{32} = 0.0$

*

$A_{33} = -10.0$

*

$B_{11} = 0.0$

*

$B_{12} = 0.0$

*

$B_{21} = 0.0$

*

$B_{22} = -14.1723156$

*

```

      B31 = 10.0
*
      B32 = 0.0
*
      A= 1.0
      B = -2.0*(A11 + A22)
      C = A11*A22 - A21*A12
      IMCHK = B**2.0 - 4.0*A*C
      IF(IMCHK.LT.0.0) IMCHK = 0.0
      LAMDA1 = (-B + SQRT(IMCHK))/2.0/A
      LAMDA2 = (-B - SQRT(IMCHK))/2.0/A
*
DERIVATIVE
NOSORT
*
*   COMPUTE INPUTS TO THE MULTIPLE LINEARIZATION MODEL.
*   RUN #1
*
      DMF = MFD - MFI
      DQD = QDD - QDI
      DNGDOT = A11*DNG + A12*DNS + A13*DE
      DNSDOT = A21*DNG + A22*DNS + A23*DE + B22*DQD
      DEDOT = A33*DE + B31*DMF
*
*
*   DYNAMIC EQUATIONS FOR MULTIPLE LINEARIZATION MODEL.
      DNG=INTGRL(0.0,DNGDOT)
      DNS=INTGRL(0.0,DNSDOT)

```



```

DE =INTGRL(0.0,DEDOT)

NGL = NGI + DNG

NSL = NSI + DNS

SORT

*

*   THE STATEMENTS IN THE PREVIOUS (DERIVATIVE) SECTION YIELD VALUES
*   OF 'NG', AND 'NS' AS CALCULATED MULTIPLE LINEARIZATION MODEL.
*   MODELS. THE STATEMENTS BELOW ARE THE DSL STATEMENTS THAT SPECIFY
*   THE VARIOUS SIMULATION SETTINGS.
*

TERMINAL

METHOD RK5

C RELERR NS = 1.E-6, NG = 1.E-6, DNS = 1.E-6, DNG = 1.E-6
C ABSERR NS = 1.E-5, NG = 1.E-5, DNS = 1.E-5, DNG = 1.E-5
CONTROL FINTIM=35.0,DELT=0.001
C PRINT .5,NS,QFPT,T4
C PRINT 0.5,NS,NSL,NSD,NG,NGL,NGD,E,WW

PRINT .35,NGD,NGL,NSD,NSL
C PRINT .5,DNG,DNS,DE
C PRINT .5,P2GI,P4GI,P2,P4,NSD,NS,NSL,NG,NGL
C PRINT .5,QD,QDM,QHPT,QFPT,QC,E,EF,MFM
* SAVE 1.0,MFM,NG,NGD,NS,NSD
SAVE .05,MFD,NGD,NSD,NGL,NSL,E,EF,QDD
*

*   RUN #1
*

* GRAPH (DE=TEK618) TIME(LO=0.0,SC=0.2,TI=.50,NI=10,UN=SEC) ...

```

```

*          NS(LO=900,SC=100,TI=1.6,NI=5,UN=RPM) ...
*          NSD(LO=900,SC=100,TI=1.6,NI=5,UN=RPM) ...
*          NSL(LO=900,SC=100,TI=1.6,NI=5,UN=RPM) ...
*          FF(LO=100,SC=10.,TI=2.,NI=4,UN='LB/HR')
*          MFM(LO=100,SC=10.,TI=2.,NI=4,UN='LB/HR')
*          NG(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM) ...
*          NGD(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM) ...
*          NGF(LO=24000,SC=1000,TI=1.1428,NI=7,UN=RPM) ...
*
*  RUN #4
*
*  GRAPH (DE=TEK618)  TIME(LO=0.0,SC=0.2,TI=.50,NI=10,UN=SEC) ...
*          NG(LO=24000,SC=1000,TI=1.33,NI=6,UN=RPM) ...
*          NGD(LO=24000,SC=1000,TI=1.33,NI=6,UN=RPM) ...
*          NS(LO=700,SC=100,TI=1.6,NI=5,UN=RPM) ...
*          NSD(LO=700,SC=100,TI=1.6,NI=5,UN=RPM) ...
*          MFM(LO=100,SC=10.,TI=1.6,NI=5,UN='LB/HR')
*
*  RUN #4
*
*  GRAPH (DE=TEK618)  TIME(LO=0.0,SC=0.2,TI=.50,NI=10,UN=SEC) ...
*          NG(LO=21000,SC=1000,TI=1.33,NI=6,UN=RPM) ...
*          NGD(LO=21000,SC=1000,TI=1.33,NI=6,UN=RPM) ...
*          NS(LO=900,SC=100,TI=2.0,NI=4,UN=RPM) ...
*          NSD(LO=900,SC=100,TI=2.0,NI=4,UN=RPM) ...
*          MFM(LO=80,SC=10.,TI=2.0,NI=4,UN='LB/HR')
*
*  RUN #1  THIS IS FOR THESIS PRESENTATION FIGURES

```

```

*
GRAPH (G2, DE=TEK618)  TIME(LO=0.0,NI=15,TI=.50,UN=SEC) ...
                        NSD(LO=250,SC=100,TI=.25,NI=36,UN=RPM) ...
*
                        NS(LO=500,SC=100,TI=.25,NI=36,UN=SEC) ...
                        NSL(LO=250,SC=100,TI=.25,NI=36,UN=RPM)
*
                        EF(LO=80.,SC=10.,TI=2.,NI=4,UN='LB/HR') ...
*
                        MFM(LO=80.,SC=10.,TI=2.,NI=4,UN='LB/HR')
GRAPH (G1, DE=TEK618)  TIME(LO=0.0,TI=.50,NI=15,UN=SEC) ...
                        NGD(LO=20000,SC=1000,TI=.25,NI=36,UN=RPM) ...
*
                        NG(LO=30000,SC=1000,TI=.25,NI=36,UN=RPM) ...
                        NGL(LO=20000,SC=1000,TI=.25,NI=36,UN=RPM)
GRAPH (G3, DE=TEK618)  TIME(LO=0.0,TI=.50,NI=15,UN='LB/HR') ...
*
                        E(LO=180.0,SC=10.,TI=.5,NI=14,UN='LB/HR') ...
*
                        EF(LO=180.,SC=10.,TI=.5,NI=14,UN='LB/HR') ...
*
                        MFD(LO=180.,SC=10.,TI=.5,NI=14,UN='LB/HR')

```

```

* GRAPH (G2, DE=TEK618) TIME, NSD, NS, NSF

```

```

* GRAPH (G1, DE=TEK618) TIME, NGD, NG, NGF

```

```

LABEL (G1)  GAS GENERATOR SPEED

```

```

LABEL (G2)  DYNAMOMETER SPEED

```

```

C

```

```

END

```

```

STOP

```

```

C          FORTRAN

```

```

C

```

APPENDIX F

COMPARISON OF "A" MATRIX COEFFICIENTS

Tables 4-9 compare the individual state space "A" matrix coefficients used in the dynamic computer simulation to those obtained from the Smoothed Dynamic Transition Map. The equation used by the dynamic simulation to calculate a particular coefficient is specified in each table by note 3. The scaling factors NSB = 1750 rpm and NGB = 26000 rpm were used to scale the variables NSL and NGL for a two variable linear regression in the forms shown.

TABLE 4

A11 MATRIX COEFFICIENT CORRELATION DATA

NS =		500 RPM	1000	1500	2000	2500
NG =	37000 RPM	-90.21	-73.87	-60.49	-49.54	-40.56
	35000	-58.71	-48.08	-39.37	-32.34	-26.40
	32000	-30.83 (-40.00)	-25.25 (-25.00)	-20.67 (-16.00)	-16.93 (-12.00)	-13.86 (-12.00)
	30000	-20.06 (-25.00)	-16.43 (-20.00)	-13.45 (-14.50)	-11.02 (-10.00)	-9.02 (-9.00)
	25000	-6.86 (-11.00)	-5.62 (-5.00)	-4.59 (-4.50)	-3.77 (-4.50)	-3.08 (-4.50)
	23000	-4.46	-3.65	-2.99	-2.45	-2.01
	22000	-3.60 (-2.00)	-2.95 (-2.20)	-2.41 (-2.20)	-1.98 (-2.10)	-1.62 (-2.00)
	20000	-2.34	-1.92	-1.57	-1.29	-1.05
	17000	-1.23	-1.00	-0.83	-0.68	-0.55
	15000	-0.80	-0.65	-0.54	-0.44	-0.36

- NOTES: 1). VALUES WITHOUT PARENTHESES ARE SIMULATION VALUES.
 2). VALUES IN PARENTHESES ARE VALUES FROM THE SMOOTHED DYNAMIC TRANSITION MAP.
 3). THE TWO VARIABLE LINEARLY REGRESSED EQUATION FOR MATRIX COEFFICIENT A11 IS:

$$A11 = -1.0 * \exp(-6.9929E-01 * (NSL/NSB) + 5.5831 * (NGL/NGB) - 3.2433)$$

TABLE 5

A12 MATRIX COEFFICIENT CORRELATION DATA

NS =		500 RPM	1000	1500	2000	2500
NG =	37000 RPM	-447.39	-198.65	-88.21	-39.17	-17.39
	35000	-241.82	-107.38	-47.68	-21.17	-9.40
	32000	-96.09 (-70.00)	-42.67 (-35.00)	-18.95 (-5.40)	-8.41 (-12.00)	-3.74 (-2.50)
	30000	-51.94 (-50.00)	-23.06 (-28.00)	-10.24 (-12.50)	-4.55 (-4.60)	-2.02 (-2.00)
	25000	-11.16 (-25.00)	-4.95 (-7.00)	-2.19 (-2.50)	-0.98 (-0.90)	-0.43 (-0.50)
	23000	-6.03	-2.68	-1.19	-0.53	-0.23
	22000	-4.43 (-3.00)	-1.97 (-0.60)	-0.87 (-0.50)	-0.39 (-0.40)	-0.17 (-0.20)
	20000	-2.39	-1.06	-0.47	-0.21	-0.09
	17000	-0.95	-0.42	-0.19	-0.08	-0.04
	15000	-0.51	-0.23	-0.10	-0.05	-0.02

- NOTES: 1). VALUES WITHOUT PARENTHESES ARE SIMULATION VALUES.
 2). VALUES IN PARENTHESES ARE VALUES FROM THE SMOOTHED DYNAMIC TRANSITION MAP.
 3). THE TWO VARIABLE LINEARLY REGRESSED EQUATION FOR MATRIX COEFFICIENT A12 IS:

$$A12 = -1.0 * \exp(-2.8415 * (NSL/NSB)) + 7.9978 * (NGL/NGB) - 4.4662$$

TABLE 6

A13 MATRIX COEFFICIENT CORRELATION DATA

NS =		500 RPM	1000	1500	2000	2500
NG =	37000 RPM	4410.37	3869.54	3395.03	2978.71	2613.44
	35000	4024.89	3531.33	3098.29	2718.36	2385.01
	32000	3508.91 (4500.00)	3078.62 (3050.00)	2701.10 (2200.00)	2369.87 (1990.00)	2079.26 (2000.00)
	30000	3202.22 (4400.00)	2809.54 (3020.00)	2465.01 (2100.00)	2162.74 (1920.00)	1897.53 (1900.00)
	25000	2547.69 (2700.00)	2235.28 (2550.00)	1961.17 (1800.00)	1720.68 (1760.00)	1509.68 (1800.00)
	23000	2325.02	2039.91	1798.76	1570.29	1377.73
	22000	2221.09 (1800.00)	1948.72 (1550.50)	1709.76 (1450.00)	1500.09 (1430.00)	1316.14 (2000.00)
	20000	2026.96	1778.39	1560.32	1368.98	1201.11
	17000	1767.11	1550.41	1360.29	1193.48	1047.13
	15000	1612.66	1414.90	1241.39	1089.17	955.61

- NOTES: 1). VALUES WITHOUT PARENTHESES ARE SIMULATION VALUES.
 2). VALUES IN PARENTHESES ARE VALUES FROM THE SMOOTHED DYNAMIC TRANSITION MAP.
 3). THE TWO VARIABLE LINEARLY REGRESSED EQUATION FOR MATRIX COEFFICIENT A13 IS:

$$A13 = -1.0 * \exp(-0.45788 * (NSL/NSB)) + 1.1890 * (NGL/NGB) + 6.8305$$

TABLE 7

A21 MATRIX COEFFICIENT CORRELATION DATA

	NS = 500 RPM	1000	1500	2000	2500
NG = 37000 RPM	1.11	0.91	0.73	0.58	0.45
35000	0.92	0.74	0.58	0.45	0.35
32000	0.67 (0.70)	0.52 (0.50)	0.39 (0.40)	0.29 (0.30)	0.22 (0.25)
30000	0.53 (0.50)	0.40 (0.40)	0.29 (0.30)	0.22 (0.20)	0.16 (0.15)
25000	0.25 (0.30)	0.18 (0.20)	0.13 (0.10)	0.09 (0.10)	0.09 (0.10)
23000	0.18	0.12	0.09	0.08	0.10
22000	0.14 (0.10)	0.09 (0.10)	0.08 (0.10)	0.08 (0.10)	0.11 (0.10)
20000	0.09	0.07	0.07	0.09	0.15
17000	0.05	0.06	0.09	0.15	0.23
15000	0.05	0.08	0.13	0.21	0.31

- NOTES: 1). VALUES WITHOUT PARENTHESES ARE SIMULATION VALUES.
 2). VALUES IN PARENTHESES ARE VALUES FROM THE SMOOTHED DYNAMIC TRANSITION MAP.
 3). THE TWO VARIABLE LINEARLY REGRESSED EQUATION FOR MATRIX COEFFICIENT A21 IS:

$$A21 = (-1.5312E-01)*((NSL/NSB)**2.0) + (-9.5315E-01)*(NSL/NSB)*(NGL/NGB) \\
(1.5745)*((NGL/NGB)**2.0) + (5.1810E-01)*(NSL/NSB) + \\
(-1.6232)*(NGL/NGB) + 4.6015E-01$$

TABLE 8

A22 MATRIX COEFFICIENT CORRELATION DATA

NS =		500 RPM	1000	1500	2000	2500
NG =	37000 RPM	-1.46	-1.44	-1.43	-1.41	-1.39
	35000	-1.36	-1.34	-1.31	-1.29	-1.28
	32000	-1.21 (-1.05)	-1.19 (-1.20)	-1.17 (-1.25)	-1.16 (-1.25)	-1.14 (-1.20)
	30000	-1.10 (-0.95)	-1.08 (-1.10)	-1.07 (-1.10)	-1.06 (-1.10)	-1.04 (-1.00)
	25000	-0.85 (-0.85)	-0.83 (-0.90)	-0.82 (-0.85)	-0.80 (-0.80)	-0.79 (-0.70)
	23000	-0.75	-0.73	-0.72	-0.70	-0.68
	22000	-0.69 (-0.75)	-0.68 (-0.80)	-0.67 (-0.70)	-0.65 (-0.60)	-0.63 (-0.50)
	20000	-0.59	-0.58	-0.57	-0.55	-0.53
	17000	-0.45	-0.43	-0.41	-0.39	-0.38
	15000	-0.34	-0.33	-0.31	-0.29	-0.28

- NOTES: 1). VALUES WITHOUT PARENTHESES ARE SIMULATION VALUES.
 2). VALUES IN PARENTHESES ARE VALUES FROM THE SMOOTHED DYNAMIC TRANSITION MAP.
 3). THE TWO VARIABLE LINEARLY REGRESSED EQUATION FOR MATRIX COEFFICIENT A22 IS:

$$A22 = (5.6875E-02)*(NSL/NSB) + (-1.3166)*(NGL/NGB) + 3.9862E-01$$

TABLE 9

A23 MATRIX COEFFICIENT CORRELATION DATA

	NS = 500 RPM	1000	1500	2000	2500
NG = 37000 RPM	1.55	2.01	2.62	3.41	4.43
35000	2.15	2.79	3.63	4.72	6.14
32000	3.51 (4.00)	4.56 (4.00)	5.93 (4.00)	7.72 (9.00)	10.04 (10.00)
30000	4.87 (4.00)	6.33 (4.00)	8.23 (9.00)	10.71 (15.00)	13.93 (15.00)
25000	11.03 (5.00)	14.35 (20.00)	18.66 (24.50)	24.27 (26.00)	31.57 (28.00)
23000	15.30	19.90	25.89	33.67	43.79
22000	18.02 (25.50)	30.49 (30.00)	39.66 (32.00)	51.58 (34.00)	67.09 (35.00)
20000	25.00	32.52	42.29	55.01	71.55
17000	40.85	53.13	69.10	89.88	116.91
15000	56.66	73.70	95.68	124.69	162.18

- NOTES: 1). VALUES WITHOUT PARENTHESES ARE SIMULATION VALUES.
 2). VALUES IN PARENTHESES ARE VALUES FROM THE SMOOTHED DYNAMIC TRANSITION MAP.
 3). THE TWO VARIABLE LINEARLY REGRESSED EQUATION FOR MATRIX COEFFICIENT A23 IS:

$$A23 = -1.0 * \exp(0.92911 * (NSL/NSB)) - 4.2459 * (NGL/NGB) + 6.2290$$

APPENDIX G

SIMULATION VS. DATA RUNS

This appendix contains the results of the three computer simulations used to validate the Smoothed Dynamic Transition Map. Figures 20-25 document the results for both NG and NS.

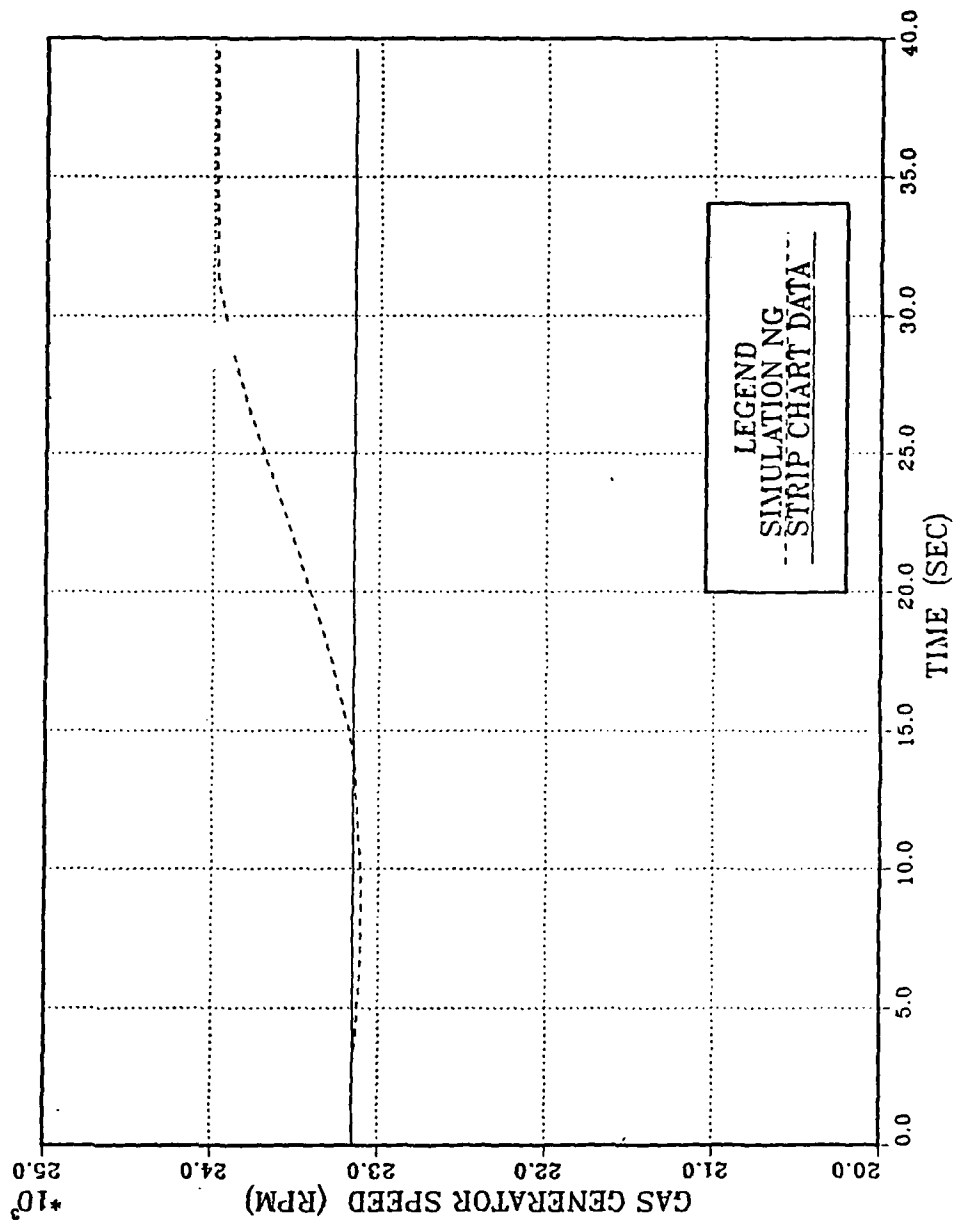


Figure 20. Simulation and Validation for Run 1 Data, NG

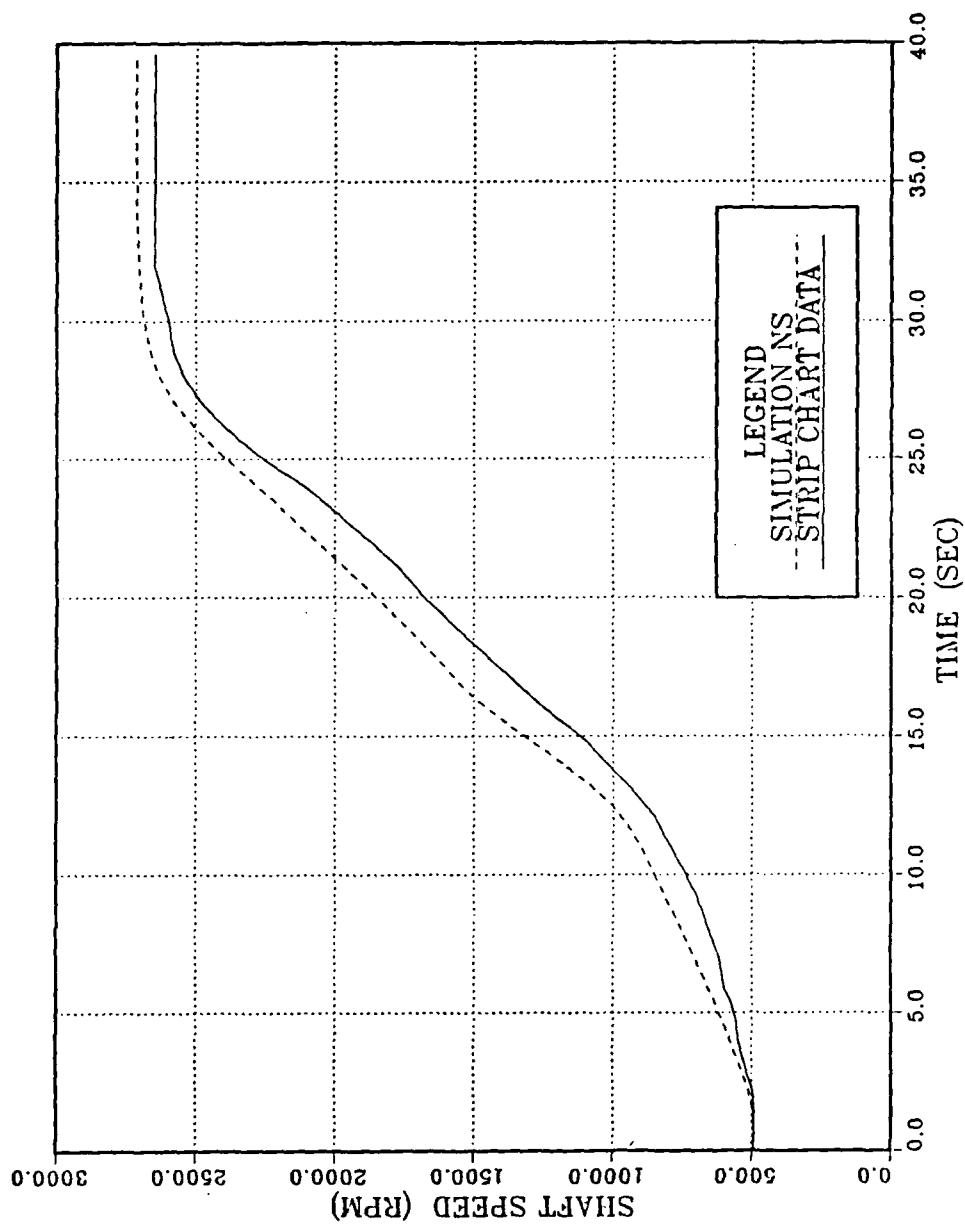


Figure 21. Simulation and Validation for Run 1 Data, NS

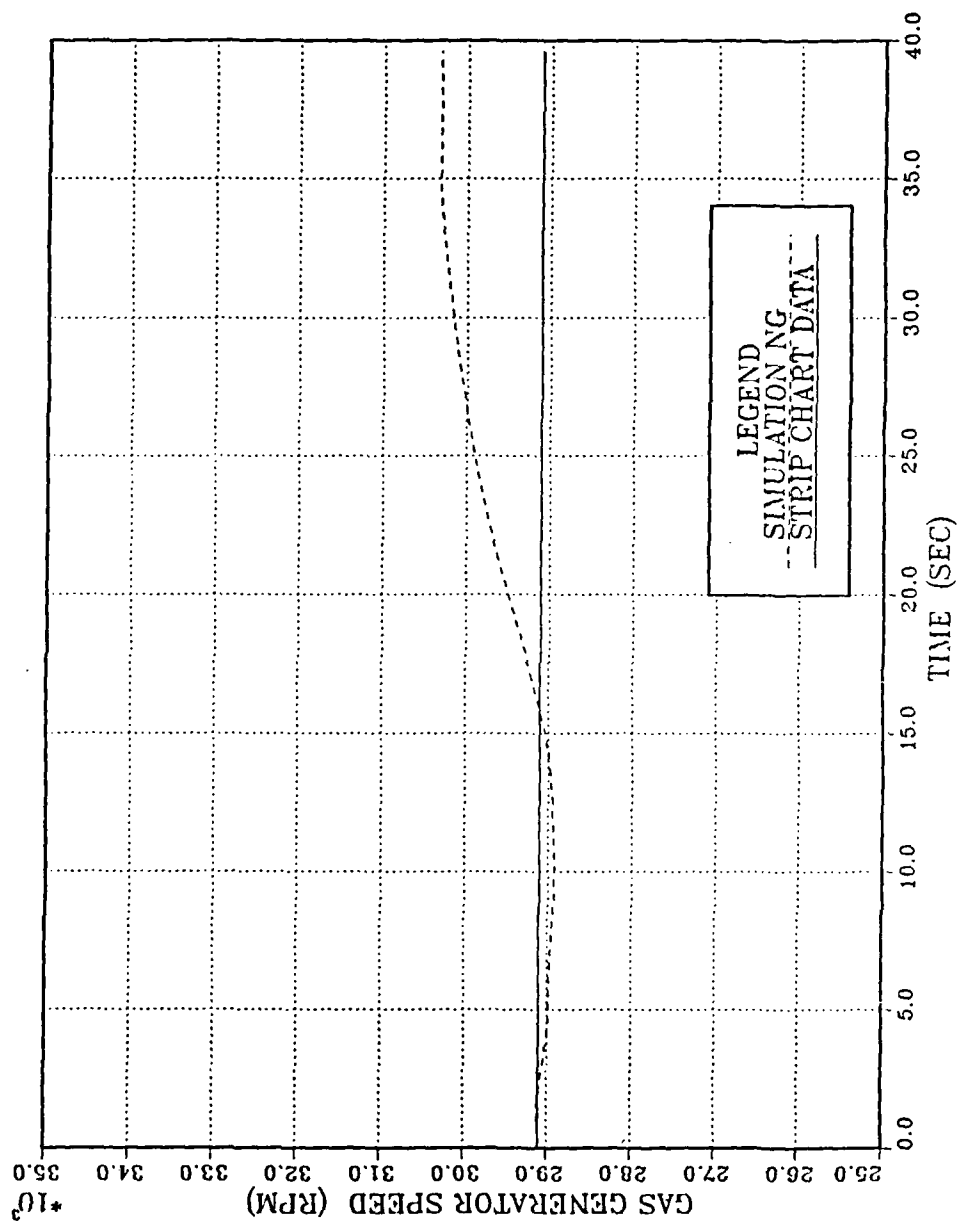


Figure 22. Simulation and Validation for Run 2 Data, NG

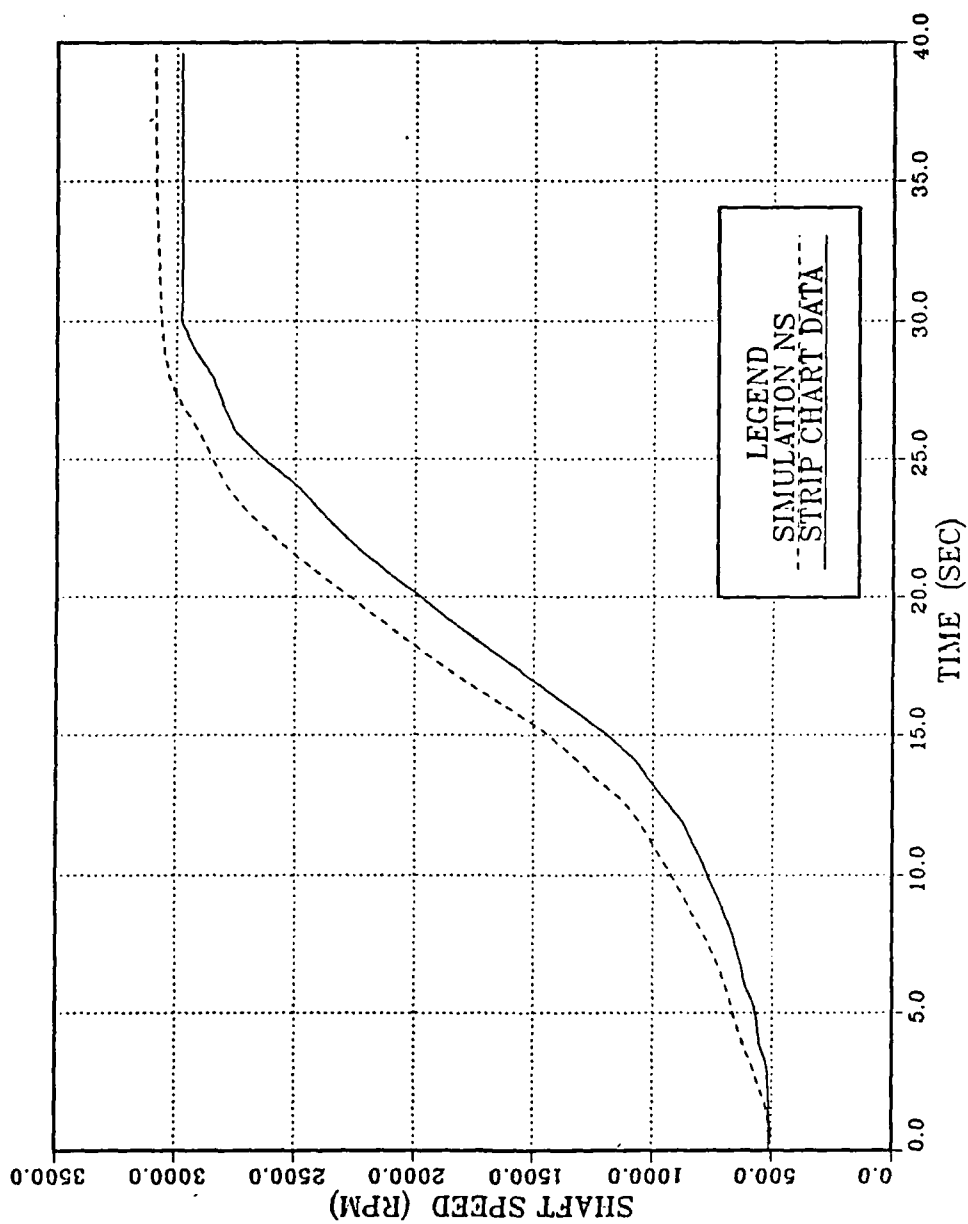


Figure 23. Simulation and Validation for Run 2, NS

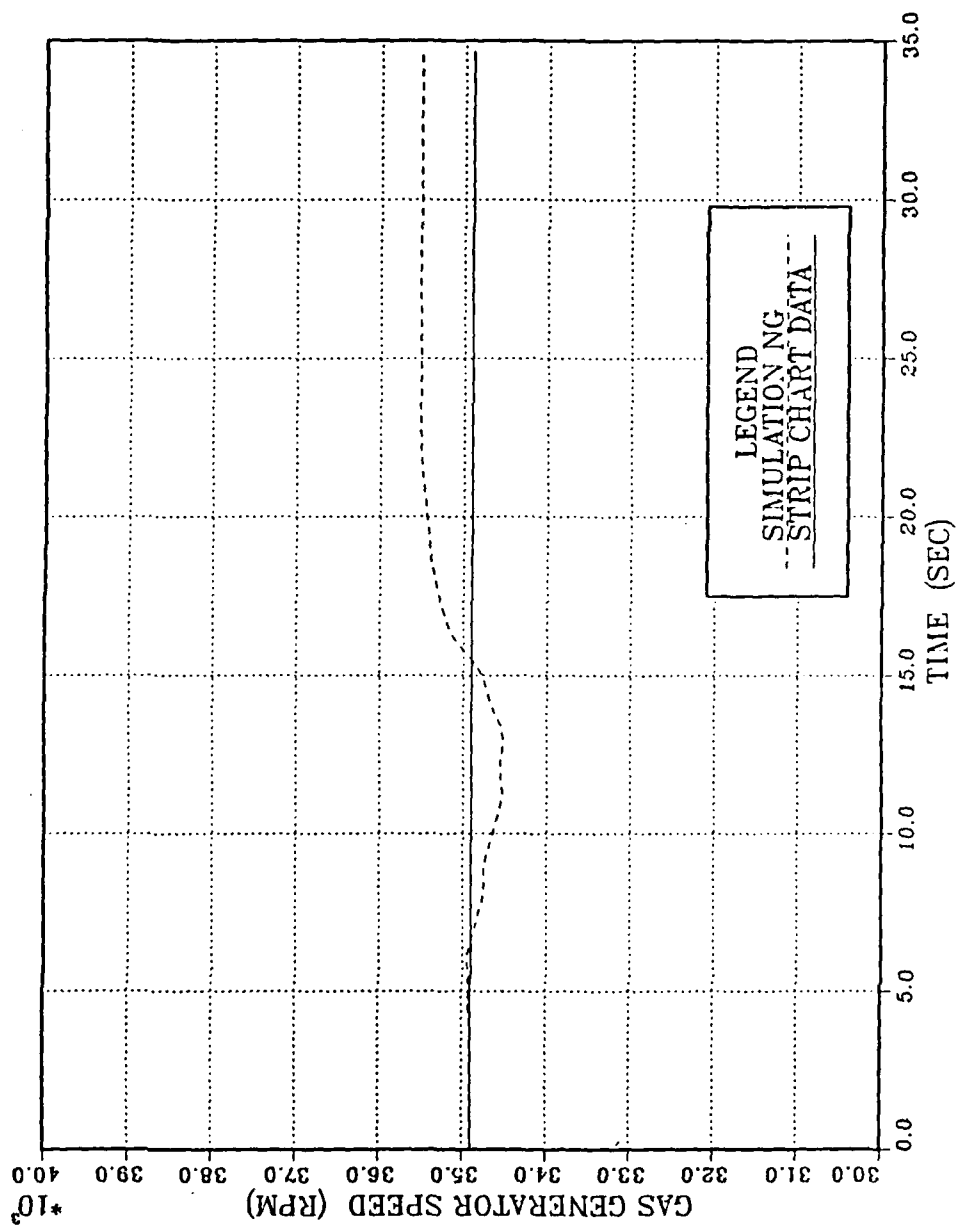


Figure 24. Simulation and Validation for Run 3, NG

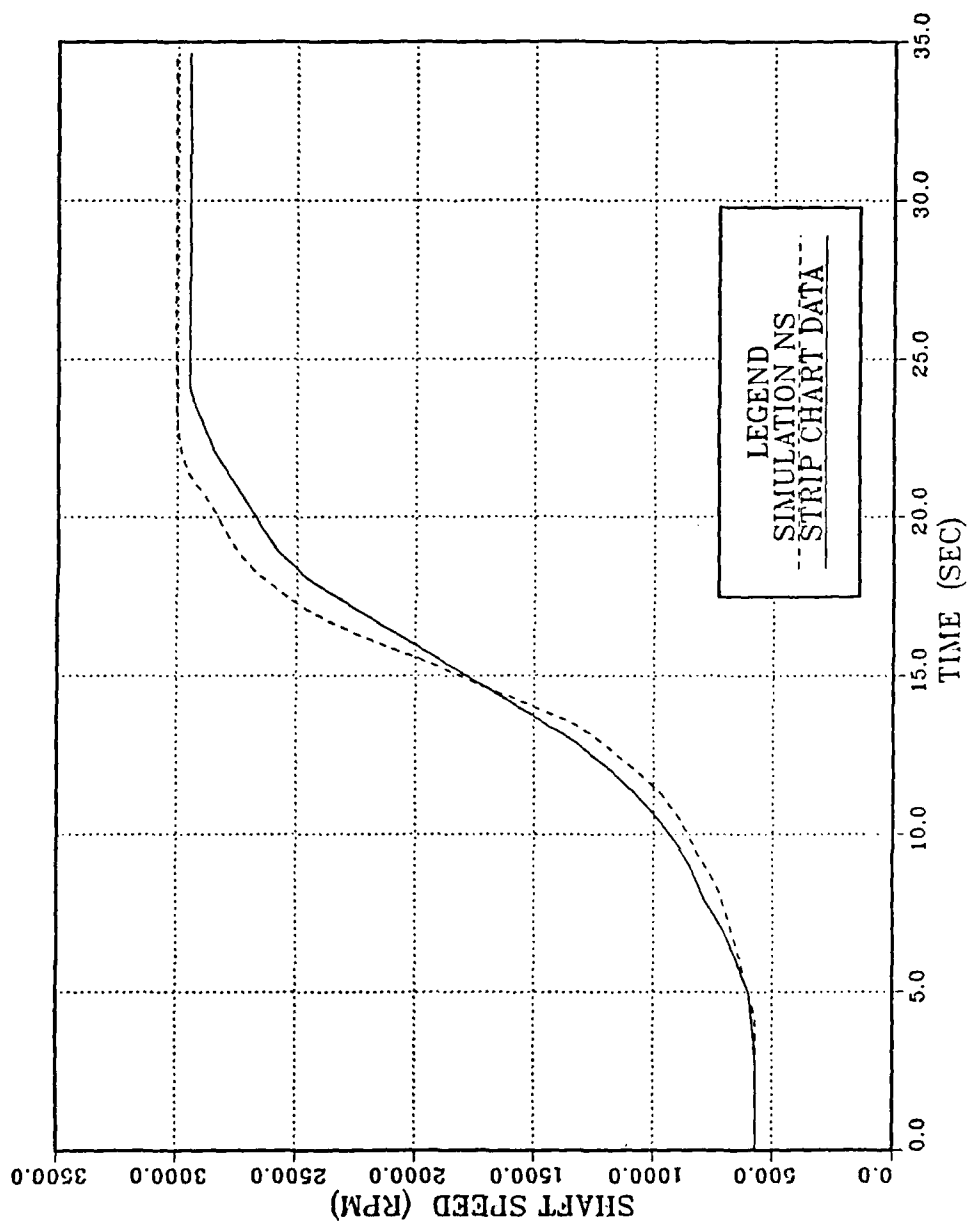


Figure 25. Simulation and Validation for Run 3, NS

LIST OF REFERENCES

1. Wheeler, D.J., "Specification of a Control System for a New Marine Gas Turbine Engine," David W. Taylor Naval Ship Research and Development Center, Annapolis, Maryland.
2. Miller, R.J. and Hackney, R.D., "F100 Multivariable Control System Engine Models/Design Criteria," AFAPL-TR-76-74, 1976.
3. Bowen, T.L., "Gas Turbine Simulation Technique for Ship Propulsion Dynamics and Control Studies," Proceedings Fifth Ship Control Systems Symposium, Vol. II, 30 October 3 November, 1978.
4. Rubis, T.J. and Harper, T.R., "The Naval Gas Turbine Propulsion Dynamics and Control Systems Research and Development Program," SNAME Transactions, Vol. 90, 1982.
5. Rubis, T.J. and Harper, T.R., "Governing Ship Propulsion Gas Turbine Engines," SNAME Transactions, Vol. 94, 1986.
6. Toney, J.H. and Houlihan, T.M., Dynamic Simulation of FFG-7 Ship Propulsion System, Master's Thesis in Mechanical Engineering, Naval Postgraduate School, Monterey, California, June 1977.
7. Xu, X. and Ni, W., "Mathematical Model of a Three-Shaft Marine Gas Turbine," International Conference on Industrial Process Modelling and Control, 6-9 June, 1985.
8. Kidd, P.T., Munro, N. and Winterbone, D.E., "Development of an Alternative Control Scheme for a Naval Propulsion Plant," Transactions of the Institute of Measurement and Control, Vol. 7, No. 1, January-March 1955.
9. Pfeil, W.H., de los Reyes, G. and Bobula, G.A., "The Application of LQR Synthesis Techniques to the Turbo-shaft Engine Control Program," AIAA/SAE/ASME 20th Joint Propulsion Conference, 11-13 June, 1984.
10. Johnson, P.N., Marine Propulsion Load Emulation, M.S. Thesis, Naval Postgraduate School, Monterey, California, June 1985.

11. Herda, V.J., Marine Gas Turbine Modelling for Modern Control Design, M.E. Thesis, Naval Postgraduate School, Monterey, California, June 1986.
12. Miller, R.L., Marine Gas Turbine Modeling for Optimal Control, M.E. Thesis, Naval Postgraduate School, Monterey, California, December 1986.
13. Clayton Manufacturing Co., Instruction Manual for the Clayton Engine Dynamometer Model 17-300-CE, El Monte, California, 1982.
14. Friedland, B., Control System Design, McGraw-Hill, 1986, pp. 338-345.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3. Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4. Professor P.F. Pucci, Code 69Pc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	1
5. Professor D.L. Smith, Code 69Sm Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	10
5. LT V.A. Stammelatti, USN Pearl Harbor Naval Shipyard Pearl Harbor, Hawaii 96860	1
6. Mr. D. Groghan Code 05R3 Naval Sea Systems Command Washington, D.C. 20362	2
7. Mr. T. Bowen Code 2721 David Taylor Naval Ship Research and Development Center Annapolis, Maryland 21402	1
8. Director of Research Administration, Code 012 Naval Postgraduate School Monterey, California 93943-5000	1

- | | |
|--|---|
| 9. LT Stephen D. Metz, USN
111 Mervine Street
Monterey, California 93940 | 1 |
| 10. LT J. Alan Davitt, USN
125 Surf Way #344
Monterey, California 93940 | 1 |
| 11. Chief of Naval Research
800 N. Quincy
Arlington, Virginia 22217-5000 | 1 |
| 12. Captain Marc Bruno, USN
PMS 375, SEMMSS
Naval Sea Systems Command
Washington, D.C. 20362-5105 | 1 |